CURSO PRÁTICO 17 DE PROGRAMAÇÃO DE COMPUTADORES

# INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 20,00

# INPUT

Vol. 2 Nº 17

## NESTE NÚMERO


**PROGRAMAÇÃO DE JOGOS**

### A CONCLUSÃO DA AVENTURA

Depois de vencer muitos obstáculos, o aventureiro defronta-se com um último perigo. O que acontecerá? Acenda a lâmpada e mate o seu coletor de impostos. O olho perdido do totem inca finalmente encontrado..... 321

**APLICAÇÕES**

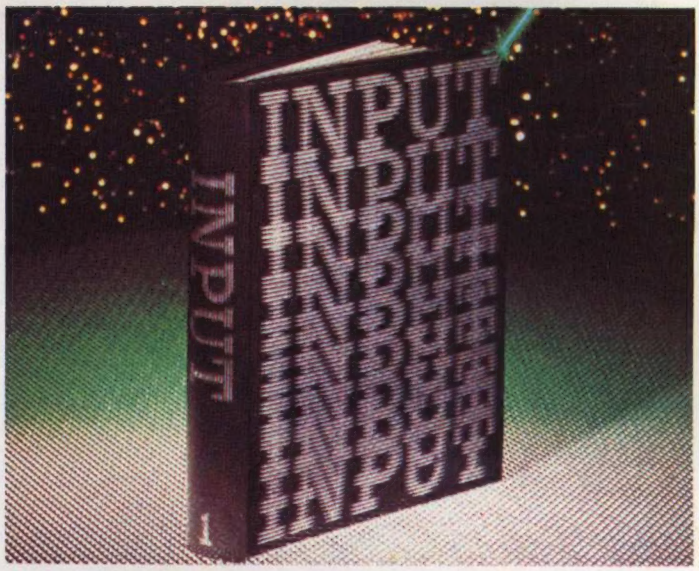
### DATILOGRAFE FRASES LONGAS

Agora que você já sabe tudo sobre o teclado, aprenda a digitar frases mais longas. Algumas regras para uma boa apresentação do texto. Use a tecla de retrocesso para corrigir os erros. Não olhe para o teclado..... 328

**PROGRAMAÇÃO BASIC**

### BÚSSOLAS E RELÓGIOS

Seno, cosseno, tangente: não é preciso conhecer trigonometria para transformar essas funções em comandos do BASIC. Os comandos **SIN**, **COS** e **TAN** e como utilizá-los no traçado de curvas, círculos e elipses. Como converter graus em radianos. Uma esfera composta de elipses..... 334


## PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

## COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: **1. Pessoalmente** — por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em São Paulo os endereços são: Rua Brigadeiro Tobias, 773 (Centro); Av. Industrial, 117 (Santo André); e, no Rio de Janeiro: Rua da Passagem, 93 (Botafogo). **2. Por carta** — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidor Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132 (Jardim Tereza) — CEP 06000 — Osasco — São Paulo. **3. Por telex** — Utilize o nº (011) 33670 ABSA. Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Ltd. — Qta. Pau Varais, Azinhaga de Fetais — 2685, Camarate — Lisboa; Tel. 257-2542 — Apartado 57 — Telex 43 069 JARLIS P.
Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na Agência do Correio. **Atenção**: Após seis meses do encerramento da coleção, os pedidos serão atendidos, dependendo da disponibilidade de estoque. **Obs.:** Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

## COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.


**Editor**
VICTOR CIVITA

**REDAÇÃO**

**Diretora Editorial:** Iara Rodrigues

**Editor chefe:** Paulo de Almeida
**Editor de texto:** Cláudio A.V. Cavalcanti
**Editor de Arte:** Eduardo Barreto
**Chefe de Arte:** Carlos Luiz Batista
**Assistentes de Arte:** Ailton Oliveira Lopes, Dilvacy M. Santos, José Maria de Oliveira, Grace A. Arruda, Monica Lenardon Corradi
**Secretária de Redação/Coordenadora:** Stefania Crema
**Secretários de Redação:** Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz
**Secretário Gráfico:** Antonio José Filho

**COLABORADORES**
**Consultor Editorial Responsável:** Dr. Renato M.E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
**Execução Editorial:** DATAQUEST Assessoria em Informática Ltda. Campinas, SP.
**Tradução:** Maria Fernanda Sabbatini
**Adaptação, programação e redação:** Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Raul Neder Porrelli
**Coordenação geral:** Rejane Felizatti Sabbatini
**Editora de Texto:** Ana Lúcia B. de Lucena
**Assistente de Arte:** Dagmar Bastos Sampaio

**COMERCIAL**
**Diretor Comercial:** Roberto Martins Silveira
**Gerente Comercial:** Flávio Ferrucio Maculan
**Gerente de Circulação:** Denise Maria Mozol

**PRODUÇÃO**
**Gerente de Produção:** João Stungis
**Coordenador de Impressão:** Atilio Roberto Bonon
**Preparador de Texto/Coordenador:** Eliel Silveira Cunha
**Preparadores de Texto:** Ana Maria Dilguerian, Antonio Francelino de Oliveira, Karina Ap. V. Grechi, Levon Yacubian, Maria Teresa Galluzzi, Paulo Felipe Mendrone
**Revisor/Coordenador:** José Maria de Assis
**Revisoras:** Conceição Aparecida Gabriel, Isabel Leite de Camargo, Ligia Aparecida Ricetto, Maria do Carmo Leme Monteiro, Maria Luiza Simões, Maria Teresa Martins Lopes.



© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.
Edição organizada pela Editora Nova Cultural Ltda.
(Artigo 15 da Lei 5 988, de 14/12/1973).


Esta obra foi composta na AM Produções Gráficas Ltda. e impressa na Divisão Gráfica da Editora Abril S.A.

# A CONCLUSÃO DA AVENTURA

- COMO SOCORRER O HERÓI
- ELIMINE O COLETOR DE IMPOSTOS
- COMO ACENDER A LÂMPADA
- INSTRUÇÕES

**Na busca incessante do olho perdido do totem inca, o herói chega à sala do trono de um reino remoto. Sairá ele com vida ou perecerá nas garras de um implacável coletor de impostos?**

Depois de muitas peripécias e lances de emoção, aproxima-se do fim nosso jogo de aventura. Antes de concluí-lo, porém, devemos acrescentar-lhe certos detalhes, como novos perigos e avisos, além de uma saída pela qual o aventureiro possa escapar com vida. Algumas instruções também precisam ser incorporadas ao jogo.

Como essas rotinas contêm detalhes específicos a esta aventura, o programa que se segue não deve ser usado em outras aventuras sem modificações. Ele apenas completará o jogo em andamento e mostrará, em termos gerais, o que é necessário para fazer um programa desse tipo. No próximo artigo, mostraremos como os princípios apresentados aqui podem ser adaptados de modo a se encaixar às suas próprias idéias.

## VOCÊ PRECISA DE AJUDA

Sempre que se encontrar diante de uma situação particularmente perigosa, o jogador (e, portanto, o personagem da aventura) pode recorrer ao computador, solicitando sugestões. Estas aparecerão na forma de mensagens impressas pela máquina em resposta ao pedido ("**AJUDAR**") do jogador. O significado de tais mensagens, assim como o momento em que elas serão impressas, depende apenas do programador. Podemos escolher, se desejarmos, não apresentar qualquer mensagem, ou fazê-las propositalmente ilusórias, ou mesmo prestar ajuda somente em alguns locais isolados.

Em nossa aventura existem vários pontos onde talvez valha a pena digitar

curtas mensagens para o jogador, advertindo-o, por exemplo, sobre o quarto escuro quando ele se encontrar em suas proximidades. Uma forma interessante de mensagem consiste em responder ao pedido de "**AJUDAR**" com um "OLHE PARA ONDE VAI", ou algo até mais enigmático.

Outra mensagem pode ser emitida às margens do rio, onde o aventureiro corre o risco de se afogar (principalmente se estiver carregando um tijolo).

Evidentemente, o número dessas mensagens pode ser muito grande. Tudo depende da vontade do programador. Suponhamos, contudo, que este queira emitir apenas um aviso, por exemplo, no rio. O primeiro passo para isso consistirá em relacionar o aviso com o número do local — número 7 — e com a variável que registra a presença do tijolo, **OB (7).**

```
3100 REM **AJUDAR**
3110 IF L<>7 OR B(2)<>-1 THEN
PRINT "DESCULPE, NAO POSSO AJUD
AR AGORA": GOTO 330
3120 PRINT "TIJOLOS SAO MUITO P
ESADOS. SEU  BRACO DEVE ESTAR D
OENDO.": GOTO 330
```

```
3100 REM **AJUDAR**
3110 IF L<>7 OR OB(2)<>-1 THEN
PRINT "DESCULPE, NAO POSSO AJUD
AR AGORA":GOTO 330
3120 PRINT "TIJOLOS SAO MUITO P
ESADOS. SEU BRACO DEVE ESTAR DO
ENDO.":GOTO 330
```

Se o aventureiro não se encontrar no rio (**L < > 7**) ou não estiver carregando o tijolo — (**OB (2) < >−1** ou **B (2) < >−1**) —, a linha 3110 imprimirá a mensagem "DESCULPE, NÃO POSSO AJUDAR AQUI". Se, pelo contrário, ele estiver transportando o tijolo e solicitar ajuda ao chegar ao rio, então aparecerá o aviso "TIJOLOS SÃO MUITO PESADOS. SEU BRAÇO DEVE ESTAR DOENDO", impresso pela linha 3120.

Nada nos impede de acrescentar uma lista completa de mensagens de ajuda nas mais diversas condições e locais de aventura.

## O COLETOR DE IMPOSTOS

Se o jogador é o "mocinho" da aventura, quem faz o papel de "bandido" é o coletor de impostos. O objetivo deste último é confiscar alguns objetos para saldar a dívida do nosso herói com o fisco. Assim, ele escolhe ao acaso um dos objetos transportados pelo jogador e o confisca, podendo levar até mesmo um tijolo como pagamento.

Se o aventureiro não estiver carregando alguma coisa quando o coletor de impostos aparecer, ele será preso e seu destino será apodrecer em uma masmorra. E assim o jogo terminará.

O papel do coletor de impostos é dar um toque de suspense ao jogo, já que ele surge de forma inesperada, independentemente do local ou de outras condições. Como em outros exemplos de acaso na programação de jogos, devemos fazer uso da função **RND**. O coletor é tratado como qualquer um dos objetos, com a diferença que sua localização é determinada ao acaso.

Aqui estão as linhas extras, para fazer o coletor de impostos aparecer.

```
320 IF  RND<(1/15)  AND  TA=0
THEN  LET B(7)=L: LET TA=1
480 IF B(7)=L AND I<>10 THEN
GOTO 1590
```

```
320 IF RND(15)=1 AND TA=0 THEN
OB(7)=L:TA=1
480 IF OB(7)=L AND I<>10 THEN 1
590
```

```
320 RN=RND(-TIME):IF INT(RND(1)
*15+1)=1 AND TA=0 THEN OB(7)=L:
TA=1
480 IF OB(7)=L AND I<>10 THEN 1
590
```

```
320  IF  INT ( RND (1) * 15 + 1
) = 1 AND TA = 0 THEN OB(7) = L
:TA = 1
480  IF OB(7) = L AND I <  > 10
 THEN 1590
```

Inicialmente, o Spectrum iguala todas essas variáveis a 0.

A linha 320 de todos os programas dá ao aventureiro uma chance em quinze de encontrar o coletor — este é o propósito do 15 no **RND**. O coletor só pode aparecer uma vez durante o jogo; portanto, você precisa da variável **TA** para saber se isso aconteceu ou não.

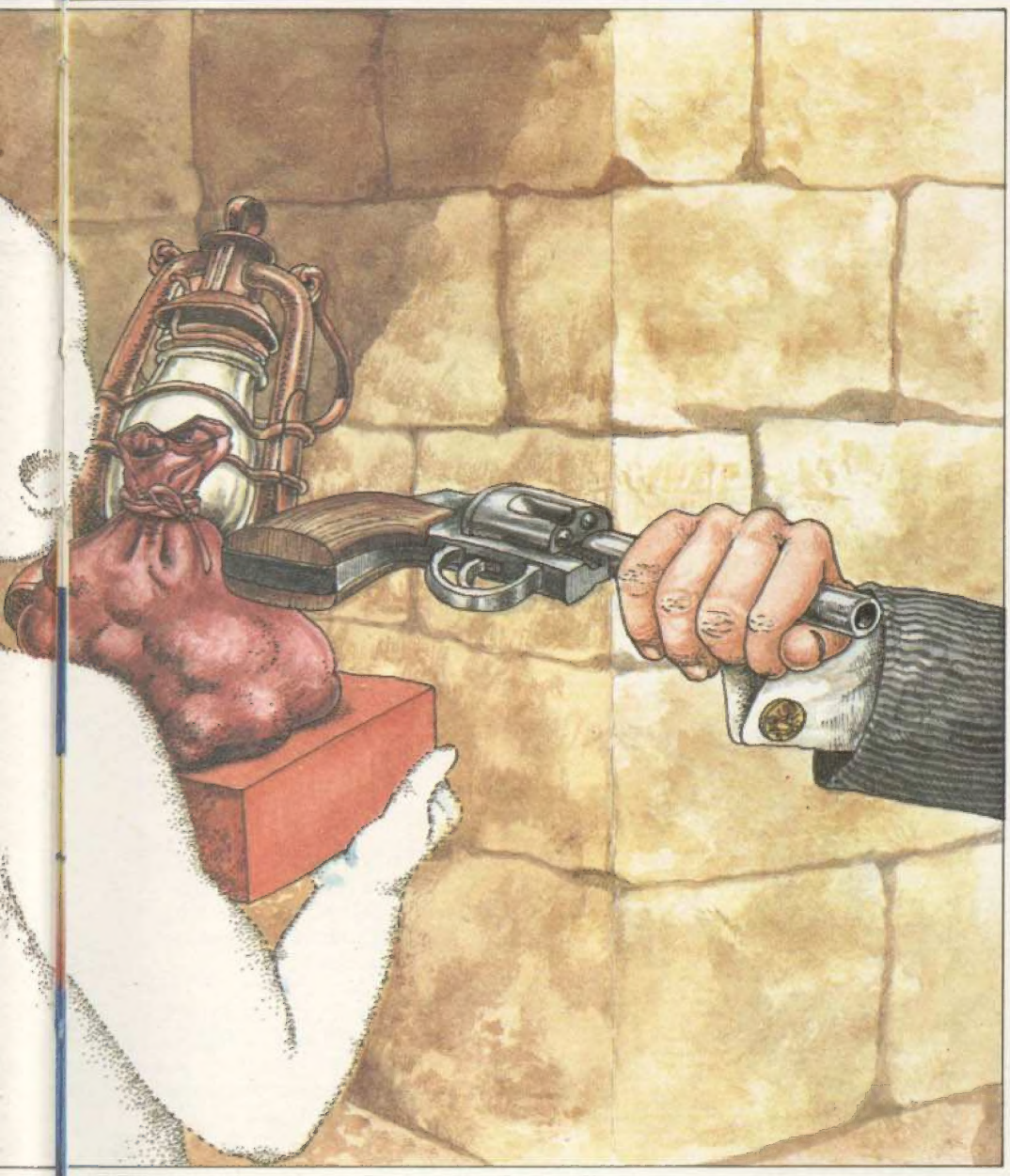
Se o número randômico for 1 (ou menor de que 1/15 no Spectrum) e o inspetor ainda não tiver aparecido, a linha

320 ajustará o valor do elemento 7 da matriz de localização de objetos — correspondente ao coletor —, de acordo com o lugar em que se encontra o jogador — **L**. O aviso de que o coletor chegou será dado como se ele fosse um objeto que tivesse sido colocado naquele local. Esse aviso nada mais é do que a descrição do objeto 7: o coletor.

A linha 480 tem a ver com o ato de eliminar o coletor, verificando se você tentou matá-lo. Se você não o fez, o programa pulará para a linha 1590.

## COMO SE LIVRAR DO COLETOR

Quando o coletor de impostos mostrar sua cara feia, o aventureiro deverá



atirar nele com o revólver que pode ser encontrado no outro lado do rio:

```
1540 REM **MATAR**
1550 IF B(4)<>-1 THEN  PRINT "C
OM O QUE?": GOTO 320
1560 IF B(7)<>L THEN  PRINT V$;
" QUEM?": GOTO 320
1570 PRINT "VOCE MATOU O ";B$(7
): LET B(7)=0: GOTO 330
```

```
1540 REM **MATAR**
1550 IF OB(4)<>-1 THEN PRINT"CO
M O QUE ?":GOTO 320
1560 IF OB(7)<>L THEN PRINT V$;
" QUEM ?":GOTO 320
1570 PRINT " VOCE MATOU O ";OB$
(7):OB(7)=0:GOTO 330
```

Essa rotina será usada quando o jogador digitar "MATAR" ou "ATIRAR". Se ele estiver sem a arma **(OB (4) < > –1)** ou **(B (4) < > –1)**, a linha 1550 perguntará "COM QUÊ?". Se o coletor não estiver e o jogador tentar matá-lo, a linha 1560 imprimirá "QUEM?". A linha 1570 comunica: "VOCÊ MATOU O COLETOR" e ajusta o elemento 7 da matriz de localização de objetos para o inspetor não mais existir.

## A VINGANÇA DO COLETOR

Aqui, o aventureiro é quem sofre nas mãos do coletor de impostos:

```
1580 REM **COLETOR DE IMPOSTOS*
1590 LET IN=0: LET B(7)=0
1600 FOR K=1 TO NB
1610 IF B(K)=-1 THEN  LET IN=IN
+1
1620 NEXT K
1630 IF IN=0 THEN  PRINT "COMO
VOCE NAO TEM NADA QUE POSSASER
CONFISCADO, ELE O PRENDE EM UMA
 MASMORRA IMUNDA": GOTO 1360
1640 LET K=INT (RND*NB)+1: IF B
(K)<>-1 THEN  GOTO 1640
1650 PRINT "ELE TOMA O ";B$(K),
"DE VOCE": LET B(K)=0: GOTO 400
```

```
1580 REM **COLETOR DE IMPOSTOS*
1590 IN=0:OB(7)=0
1600 FOR K=1 TO NB
1610 IF OB(K)=-1 THEN IN=IN+1
1620 NEXT
1630 IF IN=0 THEN PRINT "COMO V
OCE NAO TEM NADA QUE POSSASER C
ONFISCADO, ELE O PRENDE  EMUMA
MASMORRA IMUNDA":GOTO 1360
1640 K=RND(NB):IF OB(K)<>-1 THE
N 1640
1650 PRINT "ELE TOMA O ";OB$(K)
," DE VOCE":OB(K)=0:GOTO 330
```

```
1580  REM  ** COLETOR DE IMPOST
OS **
1590 IN = 0:OB(7) = 0
1600  FOR K = 1 TO NB
1610  IF OB(K) =  - 1 THEN IN =
 IN + 1
1620  NEXT
1630  IF IN = 0 THEN  PRINT "CO
MO VOCE NAO TEM NADA QUE POSSA
SER    CONFISCADO, ELE O PRENDE
 EM UMA          MASMORRA IMUNDA
": GOTO 1360
1640 K =  INT ( RND (1) * NB +
1): IF OB(K) <  >  - 1 THEN 164
0
1650  PRINT "ELE TOMA O ";OB$(K
);" DE VOCE":OB(K) = 0: GOTO 33
0
```

Como ao inspetor de taxas é permitido aparecer apenas uma vez durante a aventura, a linha 1590 ajusta o elemento 7 da matriz de localização de objetos de forma que ele não mais exista enquanto durar o programa. Esse procedimento não afeta em nada a rotina, mas evita o aparecimento do coletor no futuro. **IN** é um marcador usado para verificar se objetos estão sendo carregados.

As linhas 1600 e 1620 passam pela matriz de localização de objetos, verificando se cada objeto está sendo transportado. Qualquer objeto que estiver

sendo carregado aumenta 1 no marcador **IN**.

Se nada estiver sendo transportado, então o valor de **IN** permanecerá 0 e o aventureiro será comunicado: "COMO VOCÊ NÃO TEM NADA QUE POSSA SER CONFISCADO, ELE O PRENDERÁ EM UMA MASMORRA IMUNDA". O jogo termina e pergunta-se ao aventureiro se ele deseja uma outra jogada; isto é feito pela linha 1360.

Caso haja objetos sendo transportados, a linha 1640 pegará um deles ao acaso. Se esse objeto estiver entre os que estão sendo carregados, ele será apreendido. No entanto, se não estiver, outro objeto será escolhido a esmo e assim por diante, até que a escolha coincida com um que esteja sendo transportado.

Se um objeto adequado tiver sido selecionado, a linha 1650 comunicará ao aventureiro que objeto foi confiscado pelo coletor. O elemento da matriz de localização de objetos correspondentes será então alterado; assim, o objeto não mais existirá.

## UM POUCO DE NATAÇÃO

Esta rotina é usada quando o jogador decide atravessar o rio:

```
1400 REM **NADAR**
1410 IF L<>7 THEN  PRINT "NADAR
 ONDE ?!!": GOTO 400
1420 IF B(2)=-1 THEN  PRINT "QU
E VERGONHA, VOCE SE AFOGOU !":
GOTO 1360
1430 IF B(4)>-1 THEN  PRINT "VO
CE ACHOU UM REVOLVER": LET B(4)
=-1: GOTO 400
1440 PRINT "VOCE SE MOLHOU TODC
": GOTO 400
```

```
1400 REM **NADAR**
1410 IF L<>7 THEN PRINT " ONDE?
!!":GOTO 330
1420 IF OB(2)=-1 THEN PRINT " Q
UE VERGONHA, VOCE SE AFOGOU!":G
OTO 1360
1430 IF OB(4)>-1 THEN PRINT " V
OCE ACHOU UM REVOLVER":OB(4)=-1
:GOTO 330
1440 PRINT " VOCE SE MOLHOU TOD
O":GOTO 330
```

A linha 1410 verifica se o aventureiro está próximo ao rio. Se não estiver, o computador perguntará: "NADAR ONDE?!!" Como não existem piscinas ou praias na aventura, não há razão para escrever uma rotina de entrada para lidar com qualquer outra resposta. Nenhuma sugestão aparecerá e o jogo prosseguirá, mostrando as direções disponíveis. Caso tente atravessar o rio carregando o tijolo, o jogador morrerá — "QUE VERGONHA; VOCÊ SE AFOGOU!", comentará friamente o computador. Mas essa "morte" tem suas particularidades: depois de afundar, o jogador poderá ressuscitar, se escolher jogar novamente.

A linha 1430 verifica se o jogador carrega o revólver. Em caso negativo, o elemento da matriz de localização de objetos correspondente ao revólver será ajustado e surgirá a mensagem: "VOCÊ ENCONTROU UM REVÓLVER".

Se o aventureiro já encontrou a arma e estiver tentando atravessar o rio novamente, a linha 1440 dirá: "VOCÊ SE MOLHOU TODO".

## FINALMENTE, O OLHO PERDIDO

O jogador só poderá recuperar o tão cobiçado olho perdido do totem inca se o saco de bolas de gude tiver sido encontrado. O passo a seguir será, portanto, esvaziar o saco para o olho aparecer. Aqui está a rotina.

```
1450 REM **ESVAZIAR**
1460 IF N$<>"SACO"( TO LEN N$)
THEN  PRINT "ISTO NAO PODE SER
ESVAZIADO": GOTO 400
1470 IF B(1)<>-1 THEN  LET G=1:
 GOTO 1270
1480 PRINT "AS BOLINHAS SE ESPA
LHAM PELO CHAO": LET B(5)=L: GO
TO 370
```

```
1450 REM **ESVAZIAR**
1460 IN=INSTR("SACO",N$):IF IN<
>1 THEN PRINT " ISTO NAO PODE S
ER ESVAZIADO":GOTO 330
1470 IF OB(1)<>-1 THEN G=1:GOTO
 1270
1480 PRINT " AS BOLINHAS SE ESP
ALHAM PELO     CHAO":OB(5)=L:GOT
O 370
```

```
1450  REM  ** ESVAZIAR **
1460 IN = 0: IF N$ =  LEFT$ ("S
ACO", LEN (N$)) THEN IN = 1
1465  IF IN <  > 1 THEN  PRINT
" ISTO NAO PODE SER ESVAZIADO":
 GOTO 330
1470  IF OB(1) <  >  - 1 THEN G
 = 1: GOTO 1270
1480  PRINT " AS BOLINHAS SE ES
PALHAM PELO CHAO":OB(5) = L: GO
TO 370
```

A rotina é chamada quando o aventureiro aciona o comando "**ESVAZIAR**" alguma coisa. A linha 1460 verifica se essa coisa é o saco. Se não for (**N$ < > "SACO"**), aparecerá a mensagem "ISTO NÃO PODE SER ESVAZIADO". A linha 1470 verifica se o saco está sendo carregado (**OB (1) < > −1** ou **B (1) < > −1**). Se não estiver, em vez de emitir outra mensagem, o programa pulará para a linha 1270, que exibirá a mensagem: "VOCÊ NÃO PODE LARGAR O QUE NÃO TEM". Como o jogador não quererá "LARGAR" nada, essa linha deve ser modificada, de forma a dar uma resposta adequada. Mude-a para:

```
1270 IF B(G) < > -1 THEN PRINT
"VOCE NAO PODE ";V$;" O QUE NAO
 TEM":GOTO 330
```

```
1270 IF OB(G)<>-1 THEN PRINT "V
OCE NAO PODE ";V$;" O QUE NAO T
EM":GOTO 330
```

A variável **V$** imprimirá o verbo adequado — "LARGAR" ou "ESVAZIAR" —, conforme a situação.

Se o saco estiver sendo carregado, o programa irá para a linha 1480. A mensagem "AS BOLINHAS SE ESPALHAM PELO CHÃO" será mostrada e o elemento 5 da matriz de localização de objetos será acertado.

Não há necessidade de imprimir uma mensagem para informar o resultado porque, ao pular para a linha 370, a impressão usual de uma descrição longa poderá substituí-la. Aparecerá então no vídeo a descrição usada na matriz de descrições longas (ver linha 240).

## ACENDA A LÂMPADA

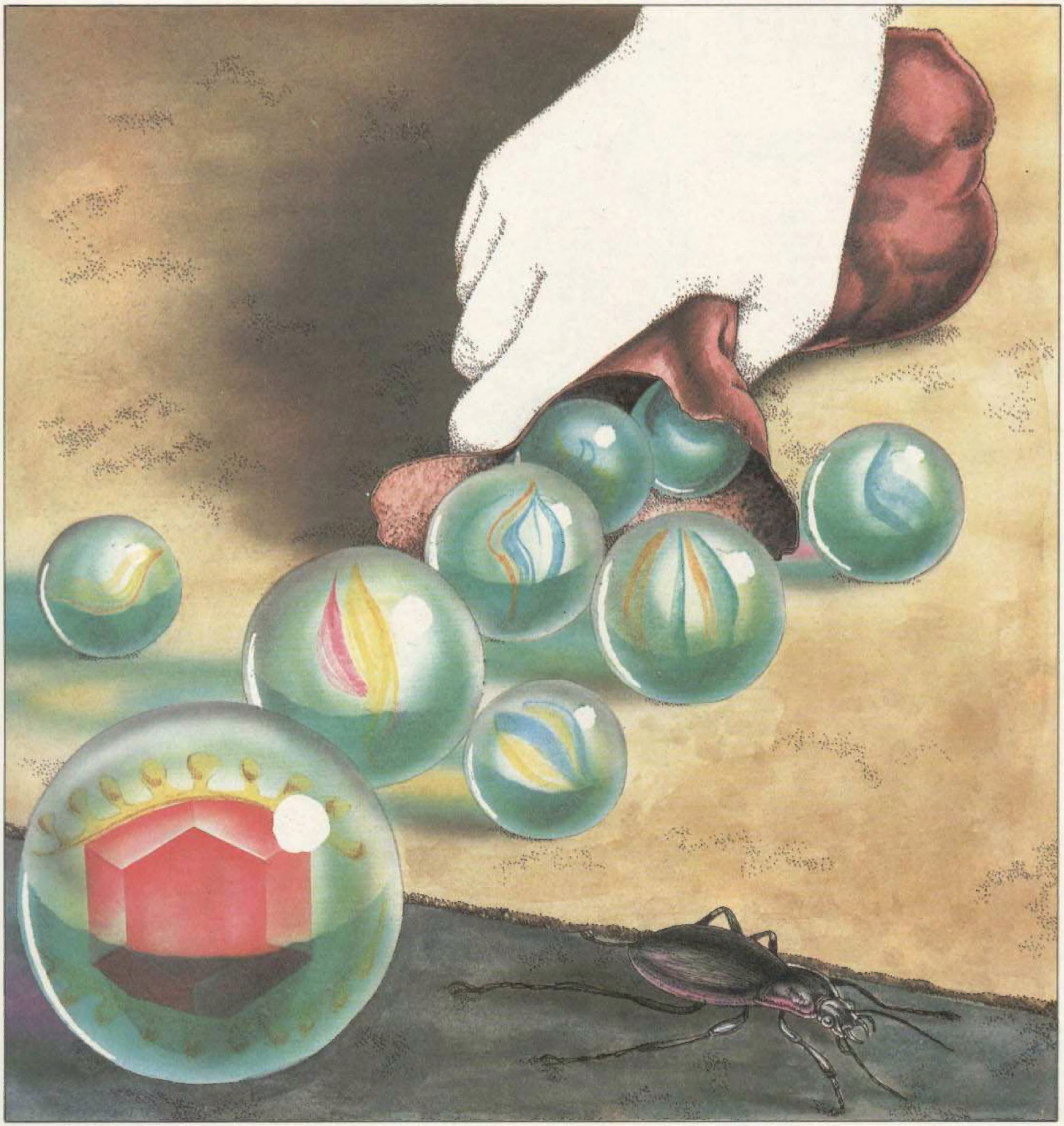
A lâmpada precisa ser acesa para que o jogador veja as saídas do quarto escuro. Se o aventureiro não estiver carregando a lanterna, a escuridão não será vencida e ele ficará preso. Esta é a rotina para acender a lâmpada.

```
1490 REM **ACENDER**
1500 IF N$<>"LAMPADA"( TO LEN N
$) THEN  PRINT "NAO PODE SER FE
ITO": GOTO 400
1510 IF B(6)<>-1 THEN  LET G=6:
 GOTO 1270
1520 IF LA=1 THEN  PRINT "JA ES
TA ACESA": GOTO 400
1530 LET LA=1: LET DA=0: PRINT
"OK": GOTO 330
```

```
1490 REM **ACENDER**
1500 IN=INSTR("LAMPADA",N$):IF
IN<>1 THEN PRINT " NAO PODE SER
 FEITO":GOTO 330
1510 IF OB(6)<>-1 THEN G=6:GOTO
 1270
1520 IF LA=1 THEN PRINT" JA EST
A ACESA":GOTO 330
1530 LA=1:PRINT "OK":GOTO 330
```

```
1490  REM  ** ACENDER **
1500 IN = 0: IF N$ =  LEFT$ ("L
AMPADA", LEN (N$)) THEN IN = 1
1505  IF IN <  > 1 THEN  PRINT
" NAO PODE SER FEITO": GOTO 330
1510  IF OB(6) <  >  - 1 THEN G
 = 6: GOTO 1270
1520  IF LA = 1 THEN  PRINT "JA
 ESTA ACESA": GOTO 330
1530 LA = 1: PRINT "OK": GOTO 3
30
```

Para que essa rotina seja requisitada, é preciso que o jogador emita a instru-



ção "**ACENDER**". A linha 1500 (1505 no Apple) é bem parecida com a linha equivalente na rotina "**ESVAZIAR**", verificando se o aventureiro escreveu a palavra "**LÂMPADA**". Caso a lanterna não esteja sendo carregada, surgirá a mensagem "VOCÊ NÃO PODE ACENDER O QUE NÃO TEM". A linha 1520 verifica se o "indicador de lâmpada acesa" — LA — está "ligado", comunicando o fato ao aventureiro. O indicador de lâmpada acesa é fixado em 1 pela linha 1530, que também imprime um "OK".

## A AVENTURA CHEGA AO FIM

Quando nosso herói entra em cena, encontra uma corrente pendurada, próxima ao trono.

O que deve ele fazer? Que tal puxar a corrente? Aqui está uma rotina que cuidará das conseqüências:

```
1300 REM **PUXAR**
1310 IF N$="CORRENTE"( TO LEN N
$) THEN  LET IN=1: IF IN=1 AND
L<>24 THEN  PRINT "NADA ACONTEC
E": GOTO 400
1320 IF IN<>1 THEN  PRINT "VOCE
 NAO PODE PUXAR ISTO !": GOTO 4
00
1330 IF B(5)<>-1 THEN  PRINT "V
OCE CAI DENTRO DO VASO E VAI EM
BORA COM A DESCARGA": GOTO 1360
1335 REM **FIM DA AVENTURA**
1340 PRINT "PARABENS ! VOCE COM
PLETOU A TAREFA."
1360 PRINT '"QUER JOGAR NOVAMEN
TE (S/N)?"
1370 LET A$=INKEY$: IF A$<>"S"
AND A$<>"N" THEN  GOTO 1370
1380 IF A$="S" THEN  RUN
1390 STOP
```

```
1300 REM **PUXAR**
1310 IN=INSTR("CORRENTE",N$):IF
 IN=1 AND L<>24 THEN PRINT " NA
DA ACONTECE":GOTO 330
1320 IF IN<>1 THEN PRINT " VOCE
 NAO PODE PUXAR ISSO!":GOTO 330
1330 IF OB(5)<>-1 THEN PRINT" V
OCE CAI DENTRO DO VASO E VAI
EMBORA COM A DESCARGA":GOTO 136
0
1340 REM **FIM DA AVENTURA**
1350 PRINT "PARABENS! VOCE COMP
LETOU A TAREFA"
1360 PRINT:PRINT" QUER JOGAR NO
VAMENTE (S/N)?"
1370 A$=INKEY$:IF A$<>"S" AND A
$<>"N" THEN 1370
1380 IF A$="S" THEN RUN
1390 END
```

```
1300  REM  ** PUXAR **
1310 IN = 0: IF N$ =  LEFT$ ("C
ORRENTE", LEN (N$)) THEN IN = 1
1315  IF IN = 1 AND L <  > 24 T
HEN  PRINT "NADA ACONTECE": GOT
O 330
1320  IF IN <  > 1 THEN  PRINT
" VOCE NAO PODE PUXAR ISSO": GO
TO 330
1330  IF OB(5) =  - 1 THEN 1340
1335  PRINT " VOCE CAI DENTRO D
O VASO E VAI EMBORA COM A DESCA
RGA": GOTO 1360
1340  REM  ** FIM DA AVENTURA *
*
1350  PRINT "PARABENS ! VOCE CO
MPLETOU A TAREFA.:? "FIMDAAVENT
URA"
```

```
1360  PRINT : PRINT " QUER JOGA
R NOVAMENTE (S/N) ?"
1370  GET A$: IF A$ <  > "S" AN
D A$ <  > "N" THEN 1370
1380  IF A$ = "S" THEN  RUN
1390  END
```

A linha 1310 considera a possibilidade de o jogador ter trazido a corrente consigo antes mesmo de puxá-la. Neste caso, ela dirá ao aventureiro que "NADA ACONTECE".

Se o herói tentar puxar qualquer outro objeto, a linha 1320 lhe dirá: "VOCÊ NÃO PODE PUXAR ISSO".

Depois disso, acontece o inesperado. Se o jogador estiver na sala do trono, não tendo porém encontrado o olho, aparecerá a mensagem "VOCÊ CAI DENTRO DO VASO E VAI EMBORA COM A DESCARGA". E o jogo termina.

Se o aventureiro der um jeito de encontrar o olho e puxou a corrente no aposento real, nenhuma dessas linhas terá efeito e ele poderá suspirar aliviado ao receber a mensagem: "PARABÉNS! VOCÊ COMPLETOU A TAREFA. FIM DA AVENTURA".

Finalmente, nas linhas 1360 e 1380, há uma opção para jogar novamente. Esta, porém, só será útil se o aventureiro tiver sido enclausurado na masmorra ou seguido com a descarga cano adentro.

## AS INSTRUÇÕES

Devemos incorporar, agora, algumas informações à rotina do jogo. Antes disso, porém, precisamos verificar o espaço restante da memória (ver página 212). Se esse espaço for pequeno, será necessário retirar todas as linhas REM — teremos então que remunerar os comandos **COSUB** que direcionam o programa para essas linhas, evitando assim mensagens de erro.

A decisão de quantas instruções serão fornecidas deve ser tomada de acordo com o espaço disponível de memória. Nelas podemos incluir até considerações quanto ao formato do vídeo do computador, o que influenciará na quantidade de detalhes a serem inseridos antes de passarmos para outra tela.

Como nossa aventura é muito simples, a rotina de instruções reduz-se a algumas linhas, contendo poucas informações. Aqui está:

```
  80 CLS : PRINT "QUER INSTRUCO
ES (S/N)?"
  90 LET A$=INKEY$: IF A$="" TH
EN  GOTO 20
  95 IF A$="S" THEN  GOSUB 6000
6000 REM **INSTRUCOES**
6010 CLS : PRINT : PRINT "  DEV
IDO A UM COLAPSO FINANCEIRO  VO
CE DEIXOU O PAIS."
6020 PRINT : PRINT "  SEUS PROB
LEMAS VAO TERMINAR      QUANDO V
OCE  ENCONTRAR  O       LEGENDA
RIO OLHO CRAVEJADO DE     BRILHA
NTES DE UM TOTEM INCA.     DEPOI
S DE FAZE-LO,VOCE TERA      QUE
ENCONTRAR A SAIDA."
6030 PRINT : PRINT "     CUIDADO
 COM O COLETOR                DE
IMPOSTOS !"
6040 PRINT AT 20,4;"PRESSIONE Q
UALQUER TECLA          PARA CONTI
NUAR"
6050 LET A$=INKEY$: IF A$="" TH
EN  GOTO 6050
6060 RETURN
```

```
10 CLS:PRINT" QUER INSTRUCOES
S/N)?"
20 A$=INKEY$:IF A$="" THEN 20
30 IF A$="S" THEN GOSUB 6000
6000 REM **INSTRUCOES**
6010 CLS:PRINT:PRINT " DEVIDO A
 UM COLAPSO FINANCEIRO  VOCE DE
IXOU O PAIS."
6020 PRINT:PRINT " SEUS PROBLEM
AS VAO TERMINAR      QUANDO VOCE
 ENCONTRAR O          LEGENDARIO
 OLHO CRAVEJADO DE     BRILHANTE
S DE UM TOTEM INCA.     DEPOIS D
E FAZE-LO VOCE TEM QUE  ENCONTR
AR A SAIDA."
6030 PRINT:PRINT"   CUIDADO  CO
M  O  COLETOR                DE
IMPOSTOS !"
6040 PRINT @451,"APERTE QUALQUE
R TECLA PARA       CONTINUAR."
6050 A$=INKEY$:IF A$="" THEN 60
50
6060 RETURN
```

```
10  HOME : PRINT "QUER INSTRUCO
ES (S/N)?"
20  GET A$: IF A$ = "" THEN 20
30  IF A$ = "S" THEN  GOSUB 600
0
6000  REM  ** INSTRUCOES **
6010  HOME : PRINT "DEVIDO A UM
 COLAPSO FINANCEIRO, VOCE "
6020  PRINT "DEIXOU O PAIS."
6025  PRINT "SEUS PROBLEMAS VAO
 TERMINAR"
6030  PRINT "QUANDO VOCE ENCONT
RAR O LEGENDARIO"
6040  PRINT "OLHO CRAVEJADO DE
BRILHANTES, DE UM "
6050  PRINT "TOTEM INCA. DEPOIS
 DE FAZE-LO, VOCE "
6060  PRINT "TERA QUE ENCONTRAR
 A SAIDA"
6070  PRINT : PRINT "CUIDADO CO
M O COLETOR DE IMPOSTOS !"
6080  PRINT : PRINT "APERTE QUA
LQUER TECLA PARA CONTINUAR"
6090  GET A$: IF A$ = "" THEN 6
090
6100  RETURN
```

Agora, grave em fita ou disco a aventura completa. A estrutura desse jogo — *O Olho Perdido do Totem Inca* — pode servir de base para suas próprias aventuras.

# DATILOGRAFE FRASES LONGAS

**Agora que você já tem certo domínio do teclado, chegou o momento de testar sua habilidade como datilógrafo, copiando frases mais longas exibidas na tela do computador.**

Se você já se sente em condições de digitar qualquer palavra ou frase a um ritmo seguro, podemos passar a uma nova etapa de nosso curso de datilografia. Até este ponto, o curso foi útil tanto para aqueles que querem apenas digitar programas quanto para os interessados em escrever cartas ou usar um editor de textos.

Nesta seção, você encontrará a oportunidade de aperfeiçoar suas habilidades, utilizando um texto mais longo. Dessa forma, além de tornar-se mais rápido, você acumulará experiência para usar o computador como uma ferramenta de escrita.

É muito importante continuar fiel à técnica de digitação usada até agora: portanto, não olhe para o teclado enquanto digita; use sempre as teclas base como referência; procure também manter o ritmo constante — não saia digitando sofregamente, a alta velocidade, mas sem segurança (a regra de ouro é iniciar a uma velocidade que possa ser mantida por algum tempo e, só então, ir acelerando aos poucos).

## COMO USAR O PROGRAMA

Ao executar o programa, o computador apresentará um menu inicial com duas opções de teste. A primeira mostra frases geradas ao acaso a partir de declarações **DATA** do programa. A segunda requer um pouco mais de trabalho. Ela permite que você use frases mais longas, mas exige que estas sejam digitadas antes. Feito isto, as frases são mostradas na tela, como no primeiro teste. Em ambos os casos você deverá datilografar o texto da maneira que ele aparece na tela; o programa lhe informará então o número de erros cometidos e a sua velocidade em palavras por minuto. É possível também escolher a maneira de proceder ao se cometer um erro (isto é, você pode usar a tecla de retrocesso para corrigi-lo ou não).

Se for escolhida a primeira opção, o engano deve ser corrigido assim que ocorre.

De outro modo, o computador aguardará até que a tecla correta seja pressionada.

O programa para o Apple é um pouco diferente do de outros micros: ele não apresenta a velocidade de digitação devido à falta de um cronômetro interno. E se o seu Apple (ou compatível) não gera caracteres minúsculos, não há problema em digitá-los todos em caracteres maiúsculos.

Ao digitar as linhas que contêm as intruções **DATA**, tome cuidado em deixar sempre um espaço no fim de cada frase, para que as palavras não fiquem justapostas quando o programa for executado.

Após algum tempo usando o programa, você verá que as frases se tornam conhecidas e monótonas, visto que há poucas variações. Tente então substituí-las por outras a seu gosto. Para fazer isso, liste as linhas **DATA** e reescreva-as com frases suas. Lembre-se de colocar entre aspas as que contêm vírgulas; faça o mesmo com as que terminam cada uma das linhas. Você evitará, assim, que elas sejam divididas em duas, ou que o espaço final seja desprezado pelo computador. Tome medidas também para que o número de frases não se altere. Se você não se sentir à vontade para fazer isto, pode variar o exercício optando por dar entrada a trechos completos de um texto, ou seja, optando pelo teste 2.

Cada passagem pode ter um máximo de 255 caracteres e o computador aceita até três frases ao mesmo tempo. Responda às perguntas que aparecerem e digite as frases de sua escolha. Depois de colocá-las na memória do computador, você poderá escolher qualquer uma delas.

Nos micros MSX, TRS-Color e Spectrum, os erros são comunicados pelo programa por intermédio de um som grave. Nos computadores da linha Apple, essa função é desempenhada por um bip.

```
10 POKE 23561,0
20 CLS
25 DIM t$(3,255): DIM t(3):
LET df=0
30 PRINT AT 7,7;"Qual teste (
1 ou 2) ?"
```

- APRENDA A DIGITAR SENTENÇAS MAIS LONGAS
- VEJA COMO O PROGRAMA FUNCIONA
- USE A TECLA DE RETROCESSO
- PARA CORRIGIR OS ERROS
- PRATIQUE COM TEXTOS MAIS LONGOS
- REGRAS PARA UMA BOA APRESENTAÇÃO DO TEXTO

```
  40 PRINT AT 10,7;"Digite '0'
  para sair"
  50 LET a$=INKEY$: IF a$<"0"
  OR a$>"2" THEN  GOTO 50
  60 IF a$="0" THEN  STOP
  70 GOSUB VAL a$*1000
  80 CLS : PRINT AT 15,4;"Palav
  ras por minuto= ";LEN c$*(INT
  ((500/(PEEK 23672+256*PEEK
  23673))*100)/100)
  90 PRINT AT 17,6;"Numero de e
  rros= ";e
 100 GOTO 30
1000 CLS : PRINT "Deseja habili
tar a tecla DELETE (s/n)?"
1010 LET a$=INKEY$: IF a$<>"n"
AND a$<>"s" THEN  GOTO 1010
1020 LET e=0: LET d=0: IF a$="s
" THEN  LET d=1
1030 CLS
1040 LET c$="": RESTORE : FOR k
=1 TO 4: LET r=INT (RND*3)+1: F
OR j=1 TO 3
1050 READ b$: IF j=r THEN  LET
c$=c$+b$
1060 NEXT j: NEXT k
1070 PRINT c$: PRINT : PRINT
1080 LET pp=0
1090 LET a$=INKEY$: IF a$="" TH
EN  GOTO 1090
1100 POKE 23672,0: POKE 23673,0
: PAUSE 0: GOTO 1120
1110 PAUSE 0
1115 LET a$=INKEY$: IF a$="" TH
EN  GOTO 1110
1120 IF a$<>c$(pp+1) AND d=0 TH
EN  GOTO 1170
1130 PRINT a$;CHR$ 95;CHR$ 8;:
LET pp=pp+1
1140 IF a$<>c$(pp) THEN  GOTO 1
170
1150 SOUND .01,30: IF pp=LEN c$
 THEN  RETURN
1160 GOTO 1110
1170 SOUND .05,-10: LET e=e+1
1180 IF d=0 THEN  GOTO 1110
1190 PAUSE 0: LET a$=INKEY$: IF
 a$<>CHR$ 12 THEN  GOTO 1190
1200 LET pp=pp-1: PRINT CHR$ 8;
CHR$ 95;" ";CHR$ 8;CHR$ 8;: GOT
O 1110
1210 GOTO 1130
1500 DATA "O cachorro manco que
 anda com tres pernas ","E fato
 que qualquer um ","Na hora cer
ta, o elefante "
1510 DATA "pode arrastar ","ser
a capaz de sentar sobre ","pode
 pular por sobre "
1520 DATA "a velha caixa esbura
cada ","a torre inclinada de Pi
sa ","qualquer uma das arvores
```

```
da fazenda "
1530 DATA "e derruba-la com um
enorme estrondo.","sem medo de
ter uma surpresa.","ate a hora
de fechar o zoologico."
2000 CLS : IF df=1 THEN  GOTO 2
015
2005 LET df=1
2010 FOR n=1 TO 3: INPUT "Intro
duza a passagem numero ";(n)' L
INE r$: LET t(n)=LEN r$: LET t$
(n)=r$: NEXT n
2015 INPUT "Que passagem voce d
eseja digitar (1 a 3)? "'p
2017 IF p>3 OR p<1 THEN  GOTO 2
015
2018 LET c$=t$(p, TO t(p))
2020 CLS : PRINT "Deseja habili
tar a tecla DELETE (s/n)?"
2030 LET a$=INKEY$: IF a$<>"s"
AND a$<>"n" THEN  GOTO 2030
2040 LET d=0: LET e=0: IF a$="s
" THEN  LET d=1
2050 CLS : GOSUB 1070: RETURN
```

```
10 CLEAR 2000
20 CLS:DIM A$(2)
30 PRINT @101,"QUAL TESTE (1 OU
 2)?"
40 PRINT @164,"PRESSIONE (0) PA
RA SAIR"
50 A$=INKEY$:IF A$<"0" OR A$>"2
" THEN 50
60 IF A$="0" THEN CLS:END
70 ON VAL(A$) GOSUB 1000,2000
80 CLS:PRINT @448,USING"PALAVRA
S POR MINUTO=###.##";LEN(C$)*50
0/TIMER
90 PRINT @480,"NUMERO DE ERROS=
";E;
100 POKE 282,255:GOTO 30
1000 CLS:PRINT " DESEJA HABILIT
AR A TECLA DE RE  TROCESSO (S/N
)?"
1010 A$=INKEY$:IF A$<>"N" AND A
$<>"S" THEN 1010
1020 E=0:D=0:IF A$="S" THEN D=1
1030 CLS:POKE 282,0
1040 C$="":RESTORE:FOR K=1 TO 4
:R=RND(3):FOR J=1 TO 3
1050 READ B$:IF J=R THEN C$=C$+
B$
1060 NEXT J,K
1070 PRINT C$
1080 PP=0
1090 A$=INKEY$:IF A$="" THEN 10
90
1100 TIMER=0:GOTO 1130
1110 A$=INKEY$:IF A$="" THEN 11
10
1120 IF A$<>MID$(C$,PP+1,1) AND
 D=0 THEN 1170
1130 PRINT @PP+256,A$:PP=PP+1
1140 IF A$<>MID$(C$,PP,1) THEN
1170
1150 SOUND 200,1:IF PP=LEN(C$)
THEN RETURN
1160 GOTO 1110
1170 SCREEN 0,1:SOUND 10,1:E=E+
1
1180 IF D=0 THEN 1110
1190 A$=INKEY$:IF A$<>CHR$(8) T
HEN 1190
1200 PP=PP-1:PRINT @256,MID$(C$
,1,PP):GOTO 1110
1210 GOTO 1130
1500 DATA O cachorro manco que
anda com tres pernas ,E fato qu
e qualquer um ,"Na hora certa,
o elefante "
1510 DATA pode arrastar ,sera c
apaz de sentar sobre ,pode pula
r por sobre
1520 DATA a velha caixa esburac
ada ,a torre inclinada de Pisa
,qualquer uma das arvores da fa
zenda
1530 DATA e derruba-la com um e
norme estrondo.,sem medo de ter
 uma surpresa.,ate a hora de fe
char o zoologico.
2000 CLS:P=0:IF A$(0)="" AND A$
```

```
(1)="" AND A$(2)="" THEN 2090
2010 PRINT" VOCE QUER USAR UMA
PASSAGEM JA  DIGITADA (S/N) ?"
2020 A$=INKEY$:IF A$<>"S" AND A
$<>"N" THEN 2020
2030 IF A$="S" THEN 2120
2040 IF A$(P)="" THEN 2090
2050 P=P+1:IF P<3 THEN 2040
2060 PRINT " AS TRES PASSAGENS
FORAM DIGITA- DAS. QUAL VOCE DE
SEJA  REESCRE- VER (1-3)?"
2070 A$=INKEY$:IF A$<"1" OR A$>
"3" THEN 2070
2080 P=VAL(A$)-1 PRINT A$:PRINT
2090 POKE 282,0:PRINT "DIGITE U
MA PASSAGEM: "
2100 LINE INPUT A$(P)
2110 GOTO 2150
2120 CLS:PRINT"QUAL PASSAGEM SE
RA USADA (1-3) ?"
2130 A$=INKEY$:IF A$<"1" OR A$>
"3" THEN 2130
2140 P=VAL(A$)-1:IF A$(P)="" TH
EN 2120
2150 CLS
2160 POKE 282,255:CLS:PRINT " D
ESEJA HABILITAR A TECLA DE RE
TROCESSO (S/N)?"
2170 A$=INKEY$:IF A$<>"S" AND A
$<>"N" THEN 2170
2180 D=0:E=0:IF A$="S" THEN D=1
2190 CLS:POKE 282,0:C$=A$(P):GO
SUB 1070:RETURN
```

```
10  HOME : DIM A$(2)
20  HTAB 8: VTAB 5: PRINT "QUAL
 TESTE? (1-2)"
30  HTAB 6: VTAB 7: PRINT "TECL
E <0> PARA TERMINAR.";
40  GET A$: IF A$ < "0" OR A$ >
 "2" THEN 40
50  IF  NOT  VAL (A$) THEN  HOM
E : END
60  ON  VAL (A$) GOSUB 1000,200
0
70  HOME : HTAB 7: VTAB 14: PRI
NT "NUMERO DE ERROS =>";E
80  GOTO 20
1000  HOME :Y = 0
1010  PRINT "VOCE QUER USAR A T
ECLA                    DE RETRO
CESSO (<-)?";
1020  GET A$: IF A$ <  > "S" AN
D A$ <  > "N" THEN 1020
1030 E = 0:D = 0: IF  ASC (A$)
= 83 THEN D = 1
1040  HOME : IF Y THEN 1080
1050 C$ = "": RESTORE : FOR K =
 1 TO 4:R =  INT ( RND (1) * 3)
 + 1: FOR J = 1 TO 3
1060  READ B$: IF J = R THEN C$
 = C$ + B$
1070  NEXT : NEXT
1080  PRINT C$:PP = 0: VTAB 12
1090  GET A$: IF A$ =  CHR$ (8)
 THEN 1090
1100  IF A$ <  >  MID$ (C$,PP +
 1,1) AND D = 0 THEN 1140
1110  PRINT A$;:PP = PP + 1
1120  IF A$ <  >  MID$ (C$,PP,1
) THEN 1140
1130  IF PP =  LEN (C$) THEN  R
ETURN
1135  GOTO 1090
1140  PRINT  CHR$ (7);:E = E +
1
1150  IF  NOT D THEN 1090
1160  GET A$: IF A$ <  >  CHR$
(8) THEN 1100
1170  PRINT A$;:PP = PP - 1: GO
TO 1090
1500  DATA  O cachorro manco qu
e anda com tres pernas ,E fato
que qualquer um ,"Na hora certa
, o elefante "
1510  DATA  pode arrastar ,sera
 capaz de sentar sobre ,"pode p
ular por sobre "
1520  DATA  a velha caixa esbur
acada ,a torre inclinada de Pis
a ,"qualquer uma das arvores da
 fazenda "
```

## MICRO DICAS

### A APRESENTAÇÃO DE UM TEXTO

Dada a facilidade com que se pode corrigir erros no teclado do computador, é possível produzir cartas limpas, sem borrões ou letras esbranquiçadas pelo uso de papel corretor.

Um documento bem montado apresenta margens de pelo menos três centímetros de ambos os lados (o texto, obviamente, deve estar centralizado no sentido vertical). Se você usar um editor de textos, as margens serão definidas automaticamente; mas, num programa seu, isso deve ser feito por intermédio dos comandos **PRINT** ou **PRINT AT**.

Tão importante quanto as margens é a distribuição correta de parágrafos. Estes fazem com que o texto fique claro e fácil de ser lido. Existem duas formas de iniciar um parágrafo: a tradicional, em que a primeira palavra começa a ser escrita alguns espaços para a direita; e a mais moderna, na qual se pula uma linha entre um parágrafo e outro, sem fazer o recuo no começo da frase. Pode-se também, evidentemente, combinar os dois métodos.

Se você tiver uma carta muito pequena para redigir, deixe um bom espaço na parte de cima do papel, antes de iniciar a impressão e/ou aumente o número de linhas entre os parágrafos. Com um pouco de prática, você produzirá textos bem centralizados e com um belo aspecto.

```
1530  DATA  e derruba-la com en
orme estrondo.,sem medo de ter
uma surpresa.,ate a hora de fec
har o zoologico.
2000  HOME :P = 0: IF A$(0) + A
$(1) + A$(2) = "" THEN 2090
2010  PRINT "VOCE QUER USAR UMA
 PASSAGEM JA DIGITADA? (S/N)";
2020  GET A$: IF A$ <  > "S" AN
D A$ <  > "N" THEN 2020
2030  IF A$ = "S" THEN 2150
2040  IF A$(P) = "" THEN 2090
2050 P = P + 1: IF P < 3 THEN 2
040
2060  PRINT : PRINT "AS TRES PA
SSAGENS JA FORAM DIGITADAS.
   QUAL VOCE DESEJA REESCREVER?
 (1-3)";
2070  GET A$: IF A$ < "1" OR A$
 > "3" THEN 2070
2080 P =  VAL (A$) - 1: PRINT A
$: PRINT
2085 A$(P) = ""
2090  PRINT : PRINT "DIGITE UMA
 PASSAGEM: "
2100  FOR I = 1 TO 255
2110  GET A$: IF A$ =  CHR$ (13
) THEN 2140
2115  IF A$ =  CHR$ (8) THEN A$
(P) =  LEFT$ (A$(P), LEN (A$(P)
) - 1): PRINT  CHR$ (8);: GOTO
2110
2120 A$(P) = A$(P) + A$
2125  HTAB 1: VTAB 5: PRINT A$(
P);
2130  NEXT
2140  GOTO 2180
2150  HOME : PRINT "QUAL PASSAG
EM SERA USADA? (1-3) ";
2160  GET A$: IF A$ < "1" OR A$
 > "3" THEN 2160
2170 P =  VAL (A$) - 1: IF A$(P
) = "" THEN 2150
2180  HOME
```

```
2190 C$ = A$(P):Y = 1: GOSUB 10
10: RETURN
```

```
10 CLEAR2000
20 R=RND(-TIME):S$="L10 O2 G"
30 CLS:DIMA$(2)
40 LOCATE 8,5:PRINT"Qual teste?
 (1 ou 2)"
50 LOCATE 6,7:PRINT"Tecle <0> p
ara terminar."
60 A$=INKEY$:IFA$<"0"ORA$>"2"TH
EN60
70 IFA$="0"THENCLS:END
80 ONVAL(A$)GOSUB1000,2000
90 CLS:LOCATE5,12:PRINT"Palavra
s por minuto => ";USING "###.##
";LEN(C$)*600/TIME
100 LOCATE7,14:PRINT"Número de
erros =>";E
110 GOTO40
1000 CLS:COLOR 15,6:Y=0
1010 PRINT"Você quer usar a tec
la de                retrocesso
 (<<)?"
1020 A$=INKEY$:IFA$<>"s"ANDA$<>
"S"ANDA$<>"n"ANDA$<>"N"THEN1020
1030 E=0:D=0:IFA$="s"ORA$="S"TH
END=1
1040 CLS:IFYTHEN1080
1050 C$="":RESTORE:FORK=1TO4:R=
INT(RND(1)*3)+1:FORJ=1TO3
1060 READB$:IFJ=RTHENC$=C$+B$
1070 NEXT:NEXT
1080 PRINTC$
1090 PP=0
1100 A$=INKEY$:IFA$=""THEN1100
1110 TIME=0:GOTO1140
1120 A$=INKEY$:IFA$=""ORA$=CHR$
(8)THEN1120
1130 IFA$<>MID$(C$,PP+1,1)ANDD=
0THEN1170
1140 VPOKE PP+201,ASC(A$):PP=PP
+1
1150 IFA$<>MID$(C$,PP,1)THEN117
0
1160 BEEP:IFPP=LEN(C$)THENCOLOR
 15,4:RETURNELSE1120
1170 PLAY S$:E=E+1
1180 IFD=0THEN1120
1190 A$=INKEY$:IFA$=""THEN1190
1200 IFA$<>CHR$(8)THEN1130
1210 PP=PP-1:VPOKEPP+201,32:GOT
O1120
1500 DATA O cachorro manco que
anda com três pernas ,É fato qu
e qualquer um ,"Na hora certa,
o elefante "
1510 DATApode arrastar ,será ca
paz de sentar sobre ,"pode pula
r por sobre "
1520 DATAa velha caixa esburaca
da ,a torre inclinada de Pisa ,
"qualquer uma das árvores da fa
zenda "
1530 DATAe derrubá-la com enorm
e estrondo.,sem medo de ter uma
 surpresa.,até a hora de fechar
 o zoológico.
2000 CLS:P=0:IFA$(0)+A$(1)+A$(2
)=""THEN2090
2010 PRINT"Você quer usar uma p
assagem já digitada? (S/N)"
2020 A$=INKEY$:IFA$<>"S"ANDA$<>
"s"ANDA$<>"n"ANDA$<>"N"THEN2020
2030 IFA$="S"ORA$="s"THEN2120
2040 IFA$(P)=""THEN2090
2050 P=P+1:IFP<3THEN2040
2060 PRINT:PRINT"As três passag
ens foram digitadas.       Qual
 você deseja reescrever? (1-3)"
2070 A$=INKEY$:IFA$<"1"ORA$>"3"
THEN2070
2080 P=VAL(A$)-1:PRINTA$:PRINT
2090 PRINT"Digite uma passagem:
 "
2100 LINEINPUTA$(P)
2110 GOTO2150
2120 CLS:PRINT"Qual passagem se
rá usada? (1-3)"
2130 A$=INKEY$:IFA$<"1"ORA$>"3"
THEN2130
2140 P=VAL(A$)-1:IFA$(P)=""THEN
2120
2150 CLS
2160 COLOR 15,12:C$=A$(P):Y=1:G
OSUB1010:RETURN
```

**Deve-se seguir alguma regra ao modificar as frases do programa?**

Isso depende de sua capacidade como datilógrafo. Se o curso inteiro foi bem assimilado, você deve ser capaz de digitar corretamente qualquer texto, seja este longo ou curto, simples ou complexo, usando os dez dedos e não apenas dois, como fazem alguns iniciantes.

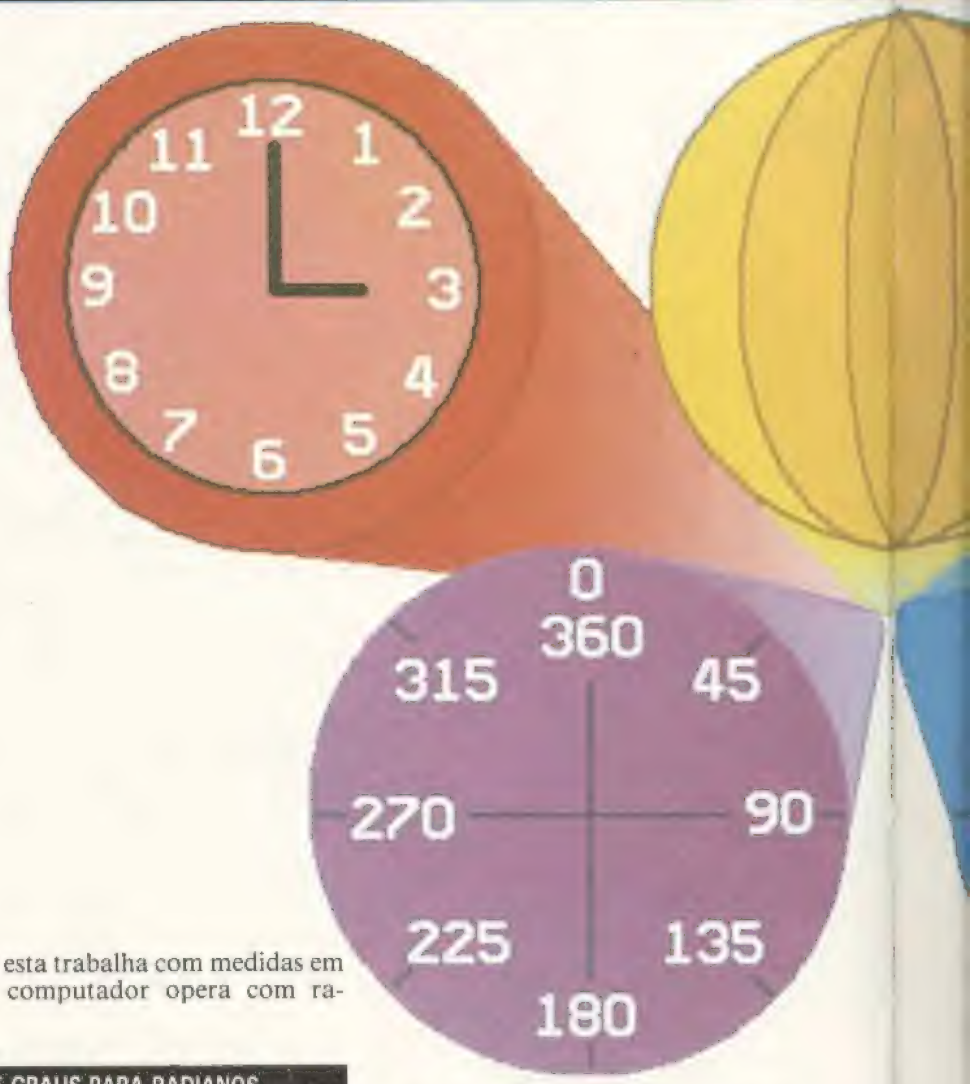
Os dedos menores, no entanto, não raro exigem mais treinamento que os outros. Para isso, certifique-se de que suas frases contêm muitos Qs, As, Zs, Ps, e Ls. É possível fazer treinamentos extra de quaisquer dedos, montando um conjunto apropriado de palavras.

É preciso, contudo, que essas palavras apareçam misturadas a outras, de modo a cobrir todo o teclado. A frase "A zebra quase fugiu do zoológico pulando a grade" é um exemplo, assim como "Fique quieto e traga-me uma caixa com doze litros de uísque". Se você quiser aumentar a destreza dos dedos menores, faça novos exercícios, inventando outras frases como essas.

O programa desta lição combina três frases para formar um texto que tenha algum sentido. Com um pouco de imaginação, você poderá conseguir o mesmo efeito com suas próprias frases.

# BÚSSOLAS E RELÓGIOS

Seno, cosseno, tangente: transformadas em comandos do BASIC, essas funções trigonométricas são essenciais quando se quer desenhar curvas, círculos e elipses no computador.




Os computadores são freqüentemente relacionados com a Matemática. Embora programar em BASIC não exija muito conhecimento nessa área, a maioria dos processos do BASIC é familiar a qualquer matemático. Contudo, muitos deles são usados não só para fazer cálculos, como também para controlar várias outras operações.

Algumas das funções matemáticas mais úteis para o programador são aquelas que calculam a relação entre ângulos e distâncias; elas são, aliás, as mais indicadas para o trabalho com gráficos.

Suponhamos, por exemplo, que queremos desenhar um relógio. Poderíamos começar pela caixa, usando o comando **CIRCLE** — menos no TRS-80 e no Apple —, mas teríamos dificuldade em posicionar corretamente os números, se calculássemos cada posição manualmente. Ora, um relógio com números em posições erradas não serviria para nada.

Felizmente, as funções matemáticas podem ser usadas para fazer esse cálculo. No relógio, a distância entre dois números é de um doze avos de volta. Essa distância pode ser calculada pelo computador: basta fornecer-lhe um número em graus ou em radianos.

Uma volta completa tem, no círculo, 360 graus, o que equivale a 2xPI radianos. No relógio, os números estão posicionados a cada 30 graus (30°), ou a cada PI/6 radianos — ou seja, a cada um doze avos de volta. O número PI (lê-se "pi") é às vezes representado pela letra grega $\pi$.

Freqüentemente usado para calcular vários aspectos de um círculo, tais como sua área e comprimento, PI vale aproximadamente 22/7 (quase 3,14). Alguns computadores já têm esse número armazenado na memória (como os micros da linha Sinclair).

É aconselhável familiarizar-se tanto com graus como com radianos, pois, se os primeiros são a medida mais comum para ângulos, é com radianos que os computadores trabalham.

Se calcularmos **SIN 30** (seno de 30) ao mesmo tempo numa calculadora e num computador, os resultados serão expressos de maneira diferente, a não ser que a calculadora esteja no modo **RAD** (radiano). Isto acontece porque, enquanto esta trabalha com medidas em graus, o computador opera com radianos.

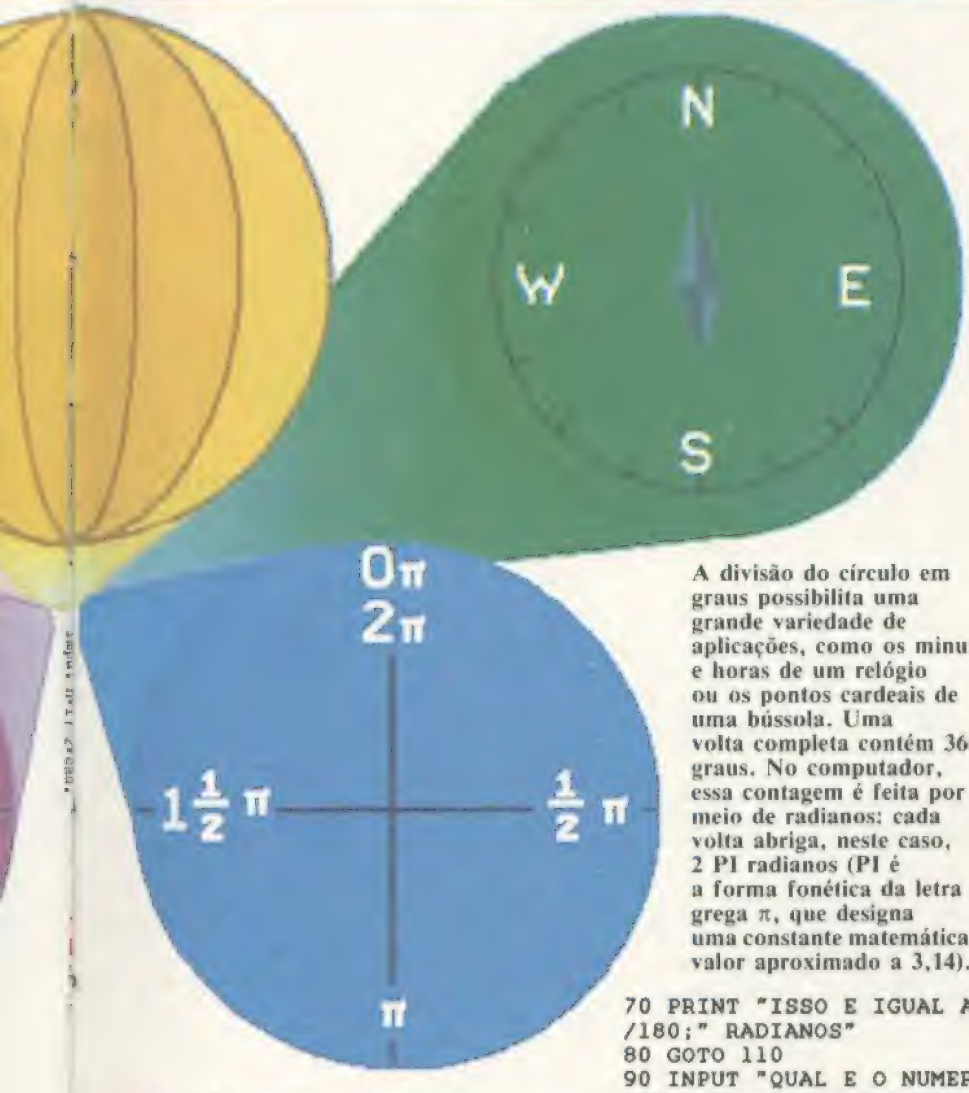
## DE GRAUS PARA RADIANOS

Para passar um número de graus para radianos basta dividi-lo por 180 e depois multiplicar o resultado por PI. Para efetuar a operação inversa (de radianos para graus), faz-se o contrário: multiplica-se o número por 180 e divide-se por PI.

O programa a seguir o ajudará a familiarizar-se com as conversões. Ele converte graus em radianos e vice-versa. Se você tiver uma calculadora, e quiser comparar os resultados, use-a da seguinte maneira: calcule o seno (comando **SIN**), o cosseno (**COS**) e a tangente (**TAN**) de um número na calculadora e anote os resultados; converta esse número em radianos no computador; faça os mesmos cálculos na calculadora e compare os resultados com os obtidos anteriormente — se tudo for feito corretamente, eles deverão ser iguais.

```
10 INPUT "DESEJA CONVERTER GR
AUS PARA     RADIANOS (1) OU
RADIANOS PARA  GRAUS (2) ?";a
20 IF a=2 THEN  GOTO 70
30 IF a<>1 THEN  GOTO 10
40 INPUT "QUAL E O NUMERO?";b
50 PRINT "E IGUAL A "; b/180*
```

- CONVERTA GRAUS EM RADIANOS
- COMO DESENHAR UMA BÚSSOLA
- COMO MEDIR ÂNGULOS NUMA BÚSSOLA
- O QUE SIGNIFICAM **SIN**, **COS** E **TAN**
- OS GRÁFICOS DE **SIN** E **COS**
- USE **SIN** E **COS** PARA DESENHAR CÍRCULOS
- UMA ESFERA FEITA DE ELIPSES




**A divisão do círculo em graus possibilita uma grande variedade de aplicações, como os minutos e horas de um relógio ou os pontos cardeais de uma bússola. Uma volta completa contém 360 graus. No computador, essa contagem é feita por meio de radianos: cada volta abriga, neste caso, 2 PI radianos (PI é a forma fonética da letra grega π, que designa uma constante matemática de valor aproximado a 3,14).**

```
PI;" RADIANOS": GOTO 90
70 INPUT "QUAL E O NUMERO?";b
80 PRINT "E IGUAL A "; b*180/
PI;" GRAUS"
90 PRINT "PRESSIONE QUALQUER
TECLA PARA   COMECAR NOVAMENT
E": PAUSE 0: CLS : GOTO 10
```

```
10 PI=4*ATN(1)
20 CLS
30 INPUT"VOCE QUER CONVERTER GR
AUS PARA  RADIANOS (1)  OU RADI
ANOS PARA  GRAUS (2) ";A
40 IF A=2 THEN GOTO 90
50 IF A<>1 THEN GOTO 20
60 INPUT "QUAL E O NUMERO ";B
70 PRINT "ISSO E IGUAL A ";B*1
/180;" RADIANOS"
80 GOTO 110
90 INPUT "QUAL E O NUMERO ";B
100 PRINT "ISSO E IGUAL A ";B*1
80/PI;" GRAUS"
110 PRINT "PRESSIONE QUALQUER T
ECLA PARA   EXECUTAR OUTRA VEZ.
"
120 A$=INKEY$:IF A$="" THEN GOT
O 120
130 GOTO 20
```

```
10 PI=4*ATN(1)
20 CLS
30 INPUT"CONVERSAO DE GRAUS PAR
A RADIANOS(1) OU DE RADIANOS PA
RA GRAUS(2) ";A
40 IF A<1 OR A>2 THEN 10
50 PRINT:INPUT"QUAL O NUMERO ";
B
60 PRINT:PRINT"ELE VALE";
70 IFA=1THEN PRINT B/180*PI;"RA
DIANOS"
80 IFA=2THEN PRINT B*180/PI;"GR
AUS"
90 PRINT:PRINT"QUALQUER TECLA P
ARA REPETIR"
100 IF INKEY$="" THEN 100
110 GOTO 20
```

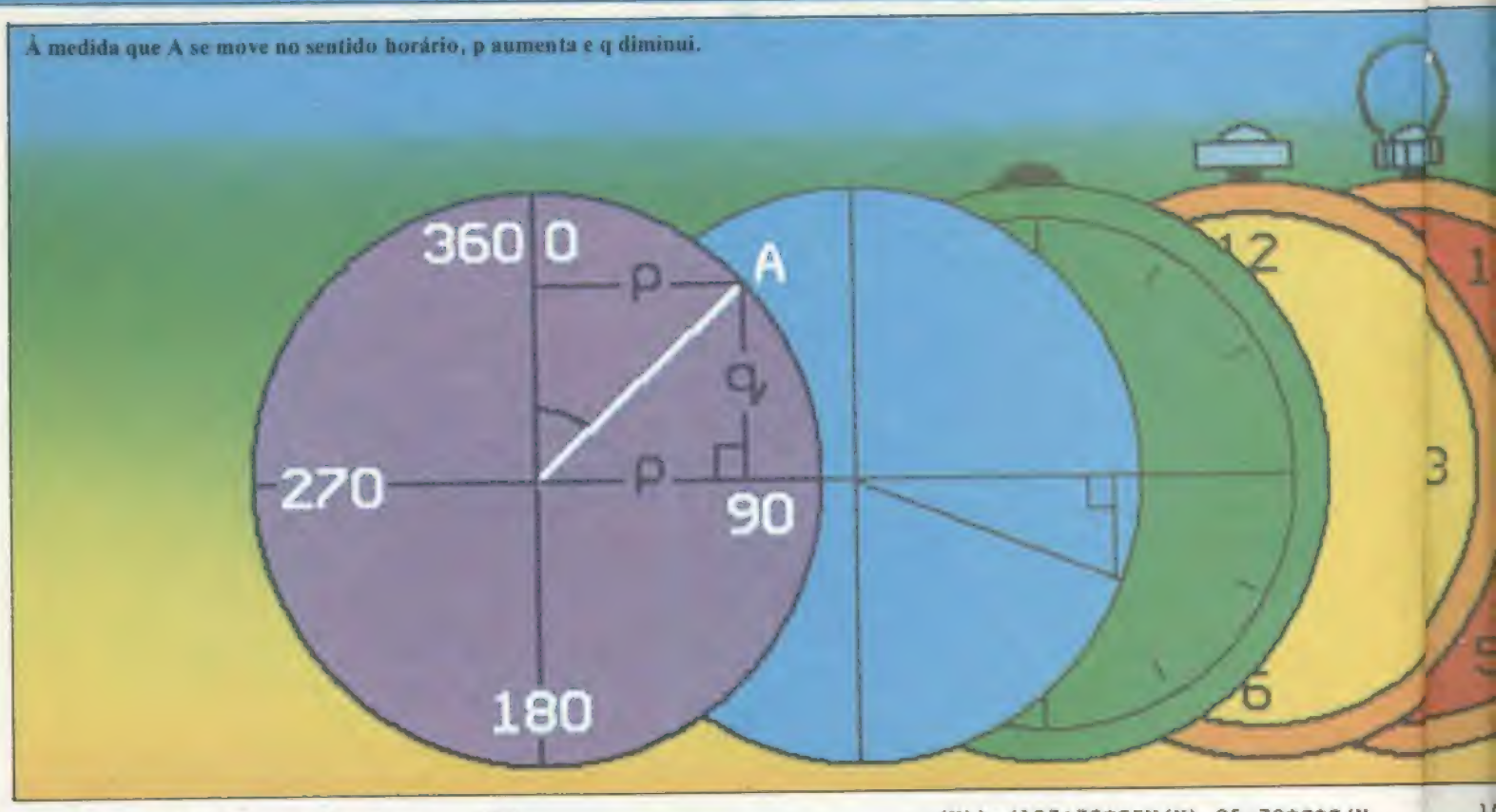
```
10 PI = 4 *  ATN (1)
20  HOME
30  INPUT "CONVERSAO DE GRAUS P
ARA RADIANOS(1) OU  DE RADIANOS
 PARA GRAUS(2) ? ";A
40  IF A <  > 1 AND A <  > 2 TH
EN 20
50  PRINT : INPUT "QUAL O NUMER
O ? ";B
60  PRINT : PRINT "ELE VALE ";
70  IF A = 1 THEN  PRINT B / 18
0 * PI;" RADIANOS"
80  IF A = 2 THEN  PRINT B * 18
0 / PI;" GRAUS"
90  PRINT : PRINT "QUALQUER TEC
LA PARA REPETIR"
100  GET A$: IF A$ = "" THEN 10
0
110  GOTO 20
```

Uma vez rodado, o programa pede por 1 ou 2, que especificam se a conversão é de graus em radianos ou de radianos em graus. Depois o programa solicita o número a ser convertido e faz aparecer o resultado na tela. Entretanto, visualizar um ângulo que é representado por um simples número pode às vezes ser muito difícil, a não ser que estejamos em condições de vê-lo desenhado. Uma representação bastante comum de todos os ângulos possíveis é encontrada nas bússolas. Poderíamos verificar isso numa bússola de verdade; mas para quê, se o computador pode mostrá-lo também?

## DESENHE UMA BÚSSOLA

Como existe uma forte relação entre círculos (ou arcos de círculos) e medidas de ângulo, podemos usar essas medidas para desenhar relógios, bússolas

À medida que A se move no sentido horário, p aumenta e q diminui.




ou quaisquer outras figuras circulares.

Os programas mencionados a seguir desenham uma bússola e posicionam as marcações de graus. O norte corresponde a 0, o leste a 90, o sul a 180 e o oeste a 270 graus. Construída a bússola, o programa desenhará qualquer ângulo que quisermos.

Somente a versão para o Sinclair Spectrum imprime números em volta da bússola. Mais adiante veremos como contornar esse problema.

Quando rodamos o programa, o computador pede por um número e traça uma linha do centro do círculo até um certo ponto, formando um ângulo de valor equivalente ao número fornecido. Isso quer dizer que, se fornecermos 90, a linha desejada partirá do centro em direção à direita. Se digitarmos 180, a linha sairá do centro e irá para baixo, representando assim um ângulo de 180 graus.

Note que o computador espera que forneçamos um número em graus para depois convertê-lo em radianos. Se olharmos para cada programa, veremos que a variável de entrada é dividida por 180 e multiplicada por PI. Isto nos poupa o trabalho de fazer a conversão manualmente.

```
10 BORDER 4: PAPER 4: INK 0:
CLS
20 CIRCLE 131,88,60
30 PLOT  131,84 :  DRAW  0,8:
 PLOT 127,88: DRAW 8,0
40 FOR a=0 TO 2*PI STEP PI/4
50 PLOT 131+55 * SIN a,88+55*
COS a: DRAW 10*SIN a,10*COS a
60 NEXT a
70 PRINT AT 2,16;0
80 FOR b=45 TO 360 STEP 45
90 PRINT AT 10-10*COS (b/180*
PI),15+10*SIN (b/180*PI);b
100 NEXT b
110 INPUT "   QUE ANGULO, EM G
RAUS, VOCE      DESEJA VER?",c
120 INK 2
130 PLOT 131,88: DRAW 45*SIN (
c/180*PI),45*COS (c/180*PI)
140 INK 0
150 INPUT " QUER RECOMECAR (s/
n)?";d$
160 IF d$="s" THEN  PLOT 131,
88: DRAW  OVER 1;45*SIN (c/180
*PI),45*COS (c/180*PI)
170 IF d$<>"s" THEN  BORDER 7:
PAPER 7: CLS : STOP
180 PLOT 131,84: DRAW 0,8:
PLOT 127,88: DRAW 8,0: GOTO
110
```

```
10 PMODE 4,1
20 PCLS
30 PI=4*ATN(1)
40 CIRCLE(127,95),80,5
50 FOR X=0 TO 2*PI STEP PI/4
60 LINE(127+72*SIN(X),95-72*COS
(X))-(127+79*SIN(X),95-79*COS(X
)),PSET
70 NEXT X
80 CLS: INPUT"QUE ANGULO VOCE Q
UER ";Z
90 SCREEN 1,1
100 X=127+60*SIN(Z*PI/180):Y=95
-60*COS(Z*PI/180)
110 LINE (127,91)-(127,99),PSET
120 LINE (123,95)-(131,95),PSET
130 LINE (127,95)-(X,Y),PSET
140 IF INKEY$="" THEN 140
150 LINE(127,95)-(X,Y),PRESET
160 GOTO 80
```

```
10 PI=4*ATN(1)
20 CLS
30 INPUT"QUE ANGULO DESEJA VER
";Z : Z=Z/180*PI
40 SCREEN2:COLOR1,15
50 CIRCLE(128,96),80,1,,,1
60 FOR X=0 TO 2*PI STEP PI/4
70 LINE(128+72*SIN(X),96-72*COS
(X))-(128+79*SIN(X),96-79*COS(X
)),1
80 NEXT X
90 LINE(128,92)-(128,100),1
100 LINE(124,96)-(132,96),1
110 FOR I=0 TO 500:NEXT
120 X=128+60*SIN(Z):Y=96-60*COS
(Z)
130 LINE(128,96)-(X,Y),1
140 FOR I=0 TO 3000:NEXT
150 GOTO 30
```

```
10  HGR
20  HCOLOR= 3: HOME
30 PI = 4 *  ATN (1)
40  FOR X = 0 TO 2 * PI STEP PI
 / 60
50  HPLOT 140 + 80 *  SIN (X),8
0 - 80 *  COS (X)
60  NEXT X
70  FOR X = 0 TO 2 * PI STEP PI
 / 4
80  HPLOT 140 + 72 *  SIN (X),8
0 - 72 *  COS (X) TO 140 + 79 *
  SIN (X),80 - 79 *  COS (X)
90  NEXT X
100  HPLOT 140,76 TO 140,84
110  HPLOT 136,80 TO 144,80
120  VTAB 23
130  INPUT "QUE ANGULO DESEJA V
ER ? ";Z:Z = Z / 180 * PI
140 X = 140 + 60 *  SIN (Z):Y =
 80 - 60 *  COS (Z)
150  HPLOT 140,80 TO X,Y
160  GET A$: IF A$ = "" THEN 16
0
170  HCOLOR= 0
180  HPLOT 140,80 TO X,Y
190  HOME : HCOLOR= 3
200  GOTO 100
```

Os programas do Apple e do TRS-Color começam ajustando o modo gráfico adequado na linha 10; depois, eles limpam a tela. Como o TRS-Color, o MSX e o Apple não têm o número PI armazenado na memória; seus programas criam uma variável, PI, com o mesmo valor do número PI.

Todos os programas, exceto o do Apple, usam depois o comando **CIRCLE** para desenhar a caixa da bússola (círculo) — linha 40 no TRS-Color, linha 20 no Spectrum e linha 50 no MSX. O Apple não possui o comando **CIRCLE**; por isso, nele o círculo é desenhado passo a passo nas linhas 40 a 60.

Os programas desenham então as marcações que completam a bússola: pequenas linhas a cada 45 graus ao redor do círculo nos ajudam a saber a exata posição de cada marcação (a versão para o Spectrum numera as posições). Tudo isso é posicionado de acordo com a abertura de cada ângulo.

A parte seguinte dos programas, menos o do MSX, pede que entremos o ângulo desejado: linha 80 para o TRS-Color, 110 para o Spectrum e 130 para o Apple e o TK-2000.

Uma pequena cruz no centro do círculo desenhado nos ajudará a verificar se os ângulos estão corretos: se entrarmos um ângulo de 90 graus, o traço lateral direito da cruz será superposto pela nova linha. A cruz é desenhada por meio de duas pequenas linhas.

Mostrado o ângulo pedido, a versão do Spectrum pergunta se queremos ver outro ângulo; a versão dos outros micros espera que apertemos qualquer tecla. Antes de desenhar um novo ângulo, o TRS-Color, o Spectrum e o Apple apagam primeiramente a linha anterior. O MSX apaga todo o desenho e o refaz depois com o ângulo novo.

Para ampliar a analogia com a bússola basta substituir as marcações de graus por N, L, S e O (ou seja, norte, leste, sul e oeste), e usar o programa como um indicador de direção. Assim, se quisermos viajar num ângulo de 270 graus, devemos entrar o valor 270; o computador desenhará então uma linha, apontando a direção desejada.

## PONTOS NUM CÍRCULO

Os programas apresentados na seção anterior usam **SIN** e **COS**, duas funções BASIC que correspondem às funções trigonométricas "seno" e "cosseno", respectivamente.

Elas se referem à posição de um ponto na circunferência em relação a dois eixos que se cortam perpendicularmente no centro do círculo. O eixo vertical é chamado de "Y" e o horizontal, de "X". Na ilustração destas páginas, o ponto A da circunferência está ligado aos dois eixos pelas linhas p e q.

Neste caso, p e q têm o mesmo comprimento. Mas, se movermos A para baixo, sobre a circunferência, perceberemos que p cresce enquanto q diminui. Quando A chegar aos 90 graus, q valerá O, enquanto p será igual ao raio da circunferência.

Usando o programa da bússola podemos visualizar melhor o que acontece. Entremos os ângulos 0, 30, 45 e 90. À medida que as linhas mudam de posição, podemos imaginar como p e q, embora não desenhadas na tela, variam de comprimento. Com uma régua poderíamos medir p e q direto da tela.

## O VALOR DE SIN E COS

Evidentemente, a relação entre p e q muda à medida que o ângulo aumenta ou diminui. Existe ainda uma relação entre o raio do círculo e as linhas p e q (quando A estiver em 90 graus, p terá o mesmo valor do raio do círculo). Assim como a anterior, essa relação entre o raio e as linhas p e q também pode ser calculada, mudando sempre que o ângulo mudar.

A relação entre o raio e p é chamada de "seno" do ângulo. A relação entre o raio e q é o "cosseno" do ângulo. Ou seja, se dividirmos as linhas p e q pelo raio, obteremos, respectivamente, o seno e o cosseno desse ângulo. Assim, se o raio for igual a 1, os valores do seno e do cosseno serão iguais aos comprimentos de p e de q.

O triângulo da ilustração abaixo foi obtido do diagrama da bússola. Um dos seus vértices é formado pelo encontro do eixo X com a linha que liga A ao centro do círculo. Uma terceira linha, que sai de A e encontra o eixo X em ângulo reto, forma o último lado do triângulo.

O seno do ângulo formado pelo encontro da linha que liga A ao centro do círculo com o eixo X é a relação entre o lado *oposto* a esse ângulo e a hipotenusa. A hipotenusa é sempre o lado oposto ao ângulo reto. Na ilustração ela é formada pela linha branca que liga A ao centro do círculo. O terceiro lado do triângulo é chamado de *adjacente*.

O cosseno também é, como vimos, uma relação entre dois lados. Existe ainda uma outra relação entre lados, conhecida como tangente do ângulo.

Em resumo, as relações são as seguintes:

seno = lado oposto/hipotenusa
cosseno = lado adjacente/hipotenusa
tangente = lado oposto/lado adjacente

As três relações podem ser calculadas pelo computador por intermédio das funções **SIN**, **COS** e **TAN**. Por exemplo, **PRINT SIN.5** coloca na tela o se-

no do ângulo .5 radianos.

Na ilustração da página 336, à medida que o ponto se move em sentido horário, partindo do topo do círculo (ângulo 0), os valores do seno e do cosseno aumentam e diminuem respectivamente, porque p e q mudam de tamanho. Depois dos 90 graus o seno começará a diminuir; quando o ponto passar pelos 180 graus tudo mudará outra vez.

Lembremos que o seno mede o quanto um ponto está *à direita* do eixo Y. Assim, quando ele atingir a metade esquerda do círculo, o seno se tornará *negativo*.

Do mesmo modo, quando o ponto estiver na metade inferior do círculo (e, portanto, abaixo do eixo X), o cosseno será negativo.

## OS GRÁFICOS DE SENO E COSSENO

Os programas a seguir mostram as mudanças do seno e do cosseno à medida que o ponto gira pelo círculo.

```
10 PLOT 0,88
20 DRAW 255,0
30 PLOT 20,0: DRAW 0,175
40 PLOT 10,158: DRAW 15,0:
PLOT 10,18: DRAW 15,0
50 PRINT AT 2,0;1;AT 20,0;-1;
AT 11,0;0;AT 20,15;180;AT 20,
29:360
60 FOR a=0 TO 2*PI STEP .06
70 PLOT 20+a*35,88+70*SIN a
80 PLOT  INK 2;20+a*35,88+70*
COS a
90 NEXT a
```

```
10 PMODE 3,1
20 PCLS
30 PI=4*ATN(1)
50 LINE(6,95)-(255,95),PSET
60 LINE(10,0)-(10,191),PSET
70 LINE(6,45)-(10,45),PSET
80 LINE(6,145)-(10,145),PSET
90 SCREEN 1,1
100 FOR X=72 TO 255 STEP 61
110 LINE(X,92)-(X,95),PSET
120 NEXT
130 FOR X=0 TO 2*PI STEP PI/123
140 PSET (123*X/PI+10,95-50*SIN
(X),3)
150 PSET (123*X/PI+10,95-50*COS
(X),2)
160 NEXT
170 GOTO 170
```

```
10 SCREEN2:COLOR1,15
20 CLS
30 PI=4*ATN(1)
40 LINE(7,96)-(256,96),1
50 LINE(11,1)-(11,192),1
60 LINE(7,46)-(11,46),1
70 LINE(7,146)-(11,146),1
80 FOR X=72 TO 256 STEP 61
90 LINE(X,93)-(X,96),1
100 NEXTX
110 FOR X=0 TO 2*PI STEP PI/123
120 PSET(123*X/PI+11,96-50*SIN(
X)),6
130 PSET(123*X/PI+11,96-50*COS(
X)),10
140 NEXTX
150 IF INKEY$="" THEN 150
160 END
```

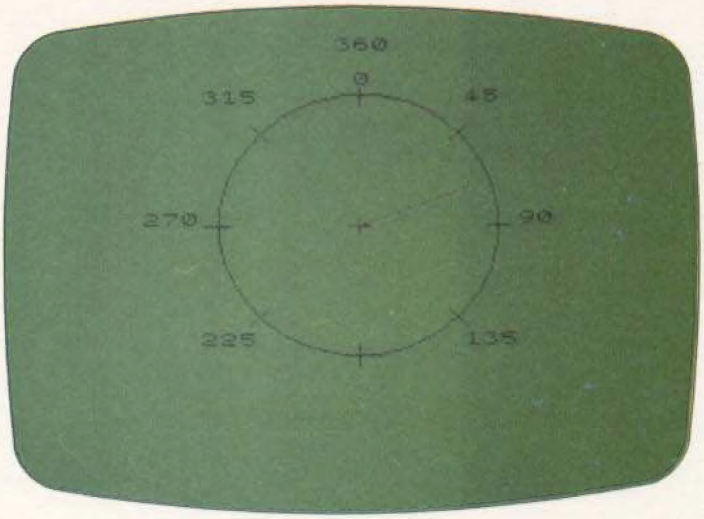
```
10  HGR2 : HCOLOR= 3
20 PI = 4 *  ATN (1)
30  HPLOT 6,95 TO 255,95
40  HPLOT 10,0 TO 10,191
50  HPLOT 6,45 TO 10,45
60  HPLOT 6,145 TO 10,145
70  FOR X = 72 TO 255 STEP 61
80  HPLOT X,92 TO X,95
90  NEXT X
100  FOR X = 0 TO 2 * PI STEP P
I / 123
110  HPLOT 123 * X / PI + 10,95
 - 50 *  SIN (X)
120  HPLOT 123 * X / PI + 10,95
 - 50 *  COS (X)
130  NEXT X
140  GET A$: IF A$ = "" THEN 14
0
150  HOME : TEXT
```

**No triângulo retângulo, a relação entre os lados é dada pelos dois ângulos não-retos. Dependendo dos lados comparados, essa relação será chamada de seno (*SIN*), cosseno (*COS*) ou tangente (*TAN*).**

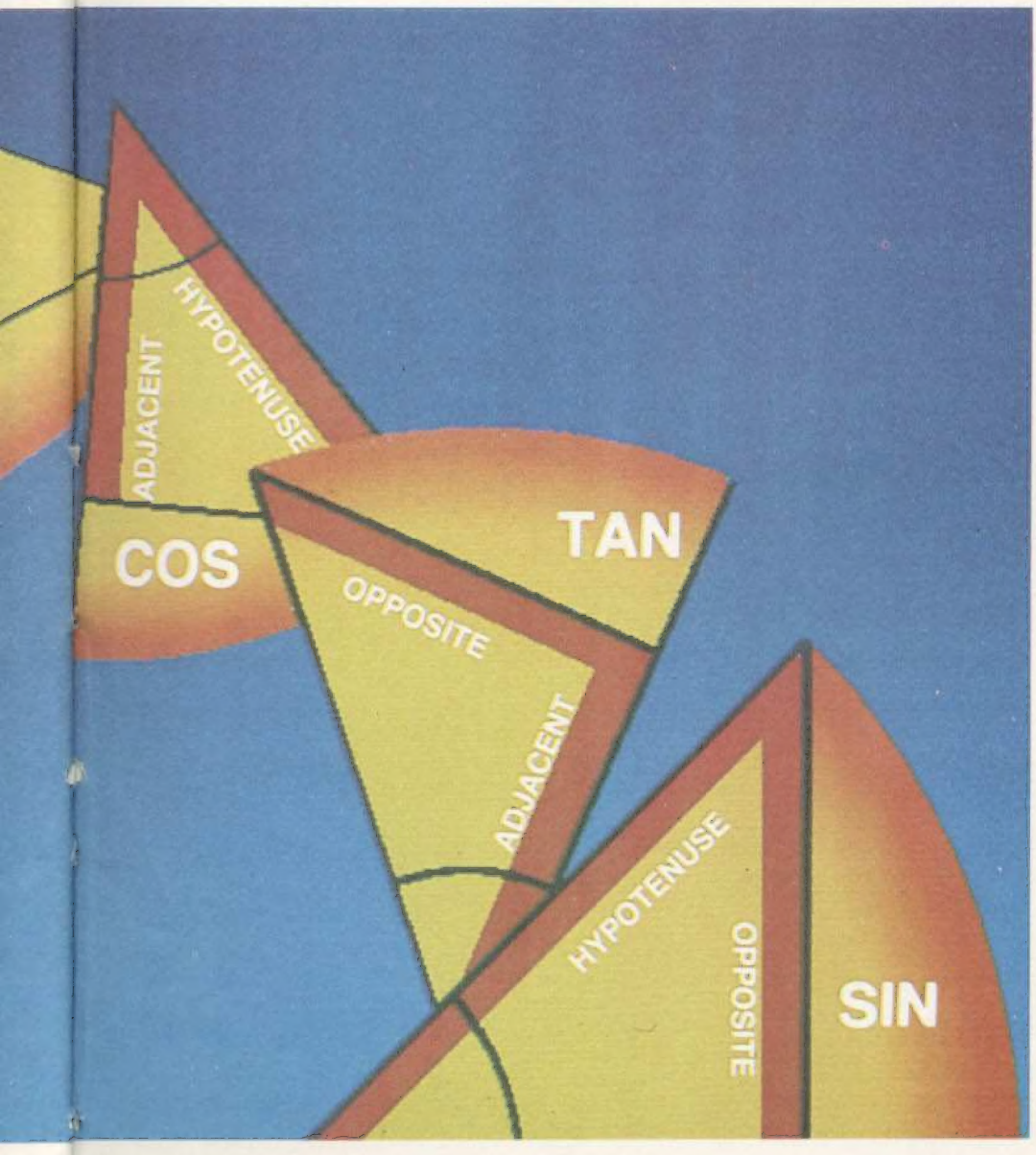
Como no programa da bússola, o MSX, o Apple e o TRS-Color começam ajustando o modo gráfico adequado (linha 10). Mais uma vez, os três criam uma variável (PI) para o valor de PI na linha 30. Depois, todos traçam os eixos para o gráfico.

Os programas desenham pequenas linhas que representam intervalos ao longo dos eixos. A versão para o Spectrum numera os traços. As marcas para o eixo Y (eixo vertical) são 1 e −1, as quais satisfazem todos os possíveis valores do seno e do cosseno. A numeração para o eixo X vai de 0 a 360 graus.

O computador desenha então os gráficos, um para o seno e outro para o cosseno. Ele faz isso usando um laço **FOR...NEXT**.




**Inventada na China por volta de 1100, a bússola magnética só se tornou conhecida na Europa um século mais tarde. Ao lado, a bússola do Sinclair Spectrum mostra um ângulo de 70 graus.**




Vimos que existem 2xPI radianos num círculo; portanto, se quisermos compreender todos os ângulos, o laço deverá ser **FOR T=0 TO 2*PI**. Nesse laço existem saltos (**STEP**) para que o computador não faça um desenho a cada mínima variação de ângulo, mas só a cada três ou quatro variações.

O **STEP** faz com que o programa seja mais rápido. Para ter uma idéia de sua importância, tente removê-lo e veja a diferença.

As linhas mais importantes para o cálculo dos valores do seno e do cosseno (140, 150 no TRS-Color; 70, 80 no TK; 120, 130 no MSX e 110, 120 no Apple e no TK-2000) parecem muito complicadas, mas na verdade são bem simples. Cada linha usa os comandos **PLOT**, **PSET** ou **DRAW** para desenhar o gráfico. As posições dos pontos são determinadas pelo seno e pelo cosseno (**SIN** e **COS**), que aparecem nessas linhas. Os outros números que os acompanham servem simplesmente para mudar os resultados de escala, a fim de ajustá-los corretamente na tela; como cada computador tem uma disposição de tela diferente, esses cálculos são diferentes para cada um.

## DESENHE CÍRCULOS

A chave para posicionar caracteres ao redor de um círculo, como no programa da bússola, consiste, como vimos, em conhecer o raio da circunferência e também os ângulos que eles formam com o eixo X.

O que precisamos é de um programa que forneça o centro e o raio do círculo e calcule as coordenadas X e Y de qualquer ângulo dado. Se entrarmos uma série de ângulos, o programa será capaz de desenhar uma série de pontos ao re-

dor do círculo escolhido. O segredo está em ensinar o computador a calcular as coordenadas X e Y, e é aqui que entram nossos velhos amigos **SIN** e **COS**.

Digite este pequeno programa:

```
20 FOR x=0 TO 2*PI STEP PI/30
30 PLOT 128+50*SIN x,88+50*
COS x
40 NEXT x
```

```
10 SCREEN2:COLOR1,15
20 CLS
30 PI=4*ATN(1)
40 FOR X=0 TO 2*PI STEP PI/45
50 PSET(128+80*SIN(X),96-80*COS
(X)),1
60 NEXTX
70 IF INKEY$="" THEN 70
80 END
```

```
10  HGR2 : HCOLOR= 3
20 PI = 4 *  ATN (1)
30  FOR Z = 0 TO 80 STEP 5
40  FOR X = 0 TO 2 * PI STEP PI
 / 45
50  HPLOT 140 + Z *  SIN (X),96
 - 80 *  COS (X)
60  NEXT X
70  NEXf Z
80  GET A$: IF A$ = "" THEN 80
90  HOME : TEXT
```

A tela exibe uma série de pontos que adotam a forma de um círculo (essa forma não é definida por uma linha contínua porque tal linha levaria muito tempo para ser desenhada).

Em cada programa, exceto no do Spectrum, o computador é ajustado para o modo gráfico adequado. Em seguida, é criado um valor para PI no TRS-Color, no MSX e no Apple. Um laço **FOR...NEXT** diz então ao computador para percorrer uma série de ângulos, que vai de 0 a 2*PI (linha 20 no TK 90X, 30 no TRS-Color, 40 no MSX e no Apple) — ou seja, diz para formar um círculo completo. O tamanho do salto (**STEP**) é o responsável pela velocidade do programa e pelo visual pontilhado.

Na linha seguinte, o computador é instruído a desenhar um ponto para cada ângulo. A posição do ponto é determinada pela fórmula:

**PLOT** raio***SIN(X)**, raio***COS(X)**

Como você deve estar lembrado, essas funções determinam as coordenadas para qualquer ângulo dado. Os outros números somados ou subtraídos nessa linha servem para ajustar o centro do círculo em relação à tela gráfica do computador. O centro da tela de alta resolução no TRS-Color está em 127,95; no Apple, está em 140,96; no MSX, está em 128,96 e no Spectrum, está em 128,88.

Tente agora alterar a posição do centro ou o tamanho do raio: isso o ajudará a entender melhor essas funções.

Existem ainda outros números que, ao serem mudados, alteram o visual do círculo. Na primeira linha do laço, o **STEP** comanda a distância entre um ponto e outro. Quanto menor o **STEP**, mais contínuo será o círculo desenhado. Se quisermos maior número de pontos no desenho, devemos diminuir o **STEP** (ou dividir PI por um número maior).

```
10 PMODE 4,1:PCLS:SCREEN 1,1
20 PI=4*ATN(1)
30 FOR X=0 TO 2*PI STEP PI/45
40 PSET (127+80*SIN(X),95-80*CO
S(X),5)
50 NEXT X
60 GOTO 60
```

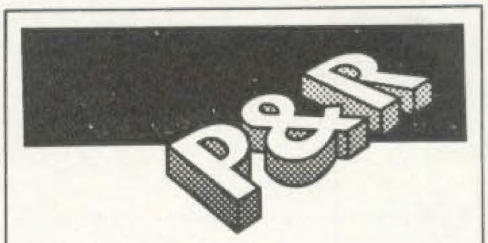
## CONSTRUA UMA ELIPSE

Uma mudança mais interessante consiste em transformar nosso círculo em uma elipse. Para tanto, basta fazermos com que o número que multiplica **SIN** seja maior ou menor que aquele que multiplica **COS**. Se o que multiplica **SIN** for maior, teremos uma elipse baixa e larga. Se for menor, a elipse será alta e estreita.

O programa a seguir desenha uma série de elipses que, juntas, criam o efeito de um globo.

```
10 FOR z=0 TO 50 STEP 5
20 FOR x=0 TO 2*PI STEP PI/15
30 PLOT 128+z*SIN x,88+50*
COS x
40 NEXT x
50 NEXT z
```

```
10 PMODE 4,1:PCLS:SCREEN 1,1
20 PI=4*ATN(1)
30 FOR Z=0 TO 80 STEP 5
40 FOR X=0 TO 2*PI STEP PI/45
50 PSET(127+Z*SIN(X),95-80*COS(
Y),5)
60 NEXT X
70 NEXT Z
80 GOTO 80
```




**Qual o tamanho do maior círculo desenhado pelo computador?**

Evidentemente, nenhum círculo desenhado pelo computador pode extravasar os limites de sua tela. De fato, o raio do maior círculo deve ser, no máximo, equivalente à metade do menor dos lados da tela. Os maiores raios para cada computador são: 88 para o Spectrum, 95 para o TRS-Color e 96 para o Apple, TK-2000 e MSX.

Entretanto, para que esses números provoquem o efeito desejado, o centro do círculo deve ser contido por eles. Se o círculo for muito grande, o Spectrum imprimirá uma mensagem de erro, enquanto os outros computadores desenharão o máximo que couber na tela.

```
10 SCREEN2:COLOR1,15
20 PI=4*ATN(1):CLS
30 FOR Z=0 TO 80 STEP 5
40 FOR X=0 TO 2*PI STEP PI/45
50 PSET(128+Z*SIN(X),96-80*COS(
X)),1
60 NEXT X
70 NEXT Z
80 IF INKEY$="" THEN 80
90 END
```

```
10  HGR2
20  HCOLOR= 3
30 PI = 4 *  ATN (1)
40  FOR X = 0 TO 2 * PI STEP PI
 / 45
50  HPLOT 140 + 80 *  SIN (X),9
6 - 80 *  COS (X)
60  NEXT X
70  GET A$: IF A$ = "" THEN 70
80  HOME : TEXT
```

Os programas do globo não passam de versões adaptadas do programa do círculo. Em vez de uma, apenas desenhamos agora uma série de elipses, usando um segundo laço **FOR...NEXT** para variar o número pelo qual a coordenada X é multiplicada. Isso faz com que o computador trace elipses de tamanhos diferentes.

Podemos fazer a elipse "crescer" o quanto quisermos, mudando o tamanho do **STEP** de Z (variável de controle).

## CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

| LINHA | FABRICANTE | MODELO |
|---|---|---|
| Apple II + | Appletronica | Thor 2010 |
| Apple II + | CCE | MC-4000 Exato |
| Apple II + | CPA | Absolutus |
| Apple II + | CPA | Polaris |
| Apple II + | Digitus | DGT-AP |
| Apple II + | Dismac | D-8100 |
| Apple II + | ENIAC | ENIAC II |
| Apple II + | Franklin | Franklin |
| Apple II + | Houston | Houston AP |
| Apple II + | Magnex | DM II |
| Apple II + | Maxitronica | MX-2001 |
| Apple II + | Maxitronica | MX-48 |
| Apple II + | Maxitronica | MX-64 |
| Apple II + | Maxitronica | Maxitronic I |
| Apple II + | Microcraft | Craf II Plus |
| Apple II + | Milmar | Apple II Plus |
| Apple II + | Milmar | Apple Master |
| Apple II + | Milmar | Apple Senior |
| Apple II + | Omega | MC-400 |
| Apple II + | Polymax | Maxxi |
| Apple II + | Polymax | Poly Plus |
| Apple II + | Spectrum | Microengenho I |
| Apple II + | Spectrum | Spectrum ed |
| Apple II + | Suporte | Venus II |
| Apple II + | Sycomig | SIC I |
| Apple II + | Unitron | AP II |
| Apple II + | Victor do Brasil | Elppa II Plus |
| Apple II + | Victor do Brasil | Elppa Jr. |
| Apple IIe | Microcraft | Craft IIe |
| Apple IIe | Microdigital | TK-3000 IIe |
| Apple IIe | Spectrum | Microengenho II |
| MSX | Gradiente | Expert GPC-1 |
| MSX | Sharp | Hotbit HB-8000 |
| Sinclair Spectrum | Microdigital | TK-90X |
| Sinclair Spectrum | Timex | Timex 2000 |
| Sinclair ZX-81 | Apply | Apply 300 |
| Sinclair ZX-81 | Engebras | AS-1000 |
| Sinclair ZX-81 | Filcres | NEZ-8000 |
| Sinclair ZX-81 | Microdigital | TK-82C |
| Sinclair ZX-81 | Microdigital | TK-83 |
| Sinclair ZX-81 | Microdigital | TK-85 |
| Sinclair ZX-81 | Prologica | CP-200 |
| Sinclair ZX-81 | Ritas | Ringo R-470 |
| Sinclair ZX-81 | Timex | Timex 1000 |
| Sinclair ZX-81 | Timex | Timex 1500 |
| TRS-80 Mod. I | Dismac | D-8000 |
| TRS-80 Mod. I | Dismac | D-8001/2 |
| TRS-80 Mod. I | LNW | LNW-80 |
| TRS-80 Mod. I | Video Genie | Video Genie I |
| TRS-80 Mod.III | Digitus | DGT-100 |
| TRS-80 Mod.III | Digitus | DGT-1000 |
| TRS-80 Mod.III | Kemitron | Naja 800 |
| TRS-80 Mod.III | Prologica | CP-300 |
| TRS-80 Mod.III | Prologica | CP-500 |
| TRS-80 Mod.III | Sysdata | Sysdata III |
| TRS-80 Mod.III | Sysdata | Sysdata Jr. |
| TRS-80 Mod.IV | Multix | MX-Compacto |
| TRS-80 Mod.IV | Sysdata | Sysdata IV |
| TRS-Color | Codimex | CS-6508 |
| TRS-Color | Dynacom | MX-1600 |
| TRS-Color | LZ | Color 64 |
| TRS-Color | Microdigital | TKS-800 |
| TRS-Color | Prologica | CP-400 |

## CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

| FABRICANTE | MODELO | PAÍS | LINHA |
|---|---|---|---|
| Appletronica | Thor 2010 | Brasil | Apple II + |
| Apply | Apply 300 | Brasil | Sinclair ZX-81 |
| CCE | MC-4000 Exato | Brasil | Apple II + |
| CPA | Absolutus | Brasil | Apple II + |
| CPA | Polaris | Brasil | Apple II + |
| Codimex | CS-6508 | Brasil | TRS-Color |
| Digitus | DGT-100 | Brasil | TRS-80 Mod.III |
| Digitus | DGT-1000 | Brasil | TRS-80 Mod.III |
| Digitus | DGT-AP | Brasil | Apple II + |
| Dismac | D-8000 | Brasil | TRS-80 Mod. I |
| Dismac | D-8001/2 | Brasil | TRS-80 Mod. I |
| Dismac | D-8100 | Brasil | Apple II + |
| Dynacom | MX-1600 | Brasil | TRS-Color |
| ENIAC | ENIAC II | Brasil | Apple II + |
| Engebras | AS-1000 | Brasil | Sinclair ZX-81 |
| Filcres | NEZ-8000 | Brasil | Sinclair ZX-81 |
| Franklin | Franklin | USA | Apple II + |
| Gradiente | Expert GPC1 | Brasil | MSX |
| Houston | Houston AP | Brasil | Apple II + |
| Kemitron | Naja 800 | Brasil | TRS-80 Mod.III |
| LNW | LNW-80 | USA | TRS-80 Mod. I |
| LZ | Color 64 | Brasil | TRS-Color |
| Magnex | DM II | Brasil | Apple II + |
| Maxitronica | MX-2001 | Brasil | Apple II + |
| Maxitronica | MX-48 | Brasil | Apple II + |
| Maxitronica | MX-64 | Brasil | Apple II + |
| Maxitronica | Maxitronic I | Brasil | Apple II + |
| Microcraft | Craft II Plus | Brasil | Apple II + |
| Microcraft | Caft IIe | Brasil | Apple IIe |
| Microdigital | TK-3000 IIe | Brasil | Apple IIe |
| Microdigital | TK-82C | Brasil | Sinclair ZX-81 |
| Microdigital | TK-83 | Brasil | Sinclair ZX-81 |
| Microdigital | TK-85 | Brasil | Sinclair ZX-81 |
| Microdigital | TK-90X | Brasil | Sinclair Spectrum |
| Microdigital | TKS-800 | Brasil | TRS-Color |
| Milmar | Apple II Plus | Brasil | Apple II + |
| Milmar | Apple Master | Brasil | Apple II + |
| Milmar | Apple Senior | Brasil | Apple II + |
| Multix | MX-Compacto | Brasil | TRS-80 Mod.IV |
| Omega | MC-400 | Brasil | Apple II + |
| Polymax | Maxxi | Brasil | Apple II + |
| Polymax | Poly Plus | Brasil | Apple II + |
| Prologica | CP-200 | Brasil | Sinclair ZX-81 |
| Prologica | CP-300 | Brasil | TRS-80 Mod.III |
| Prologica | CP-400 | Brasil | TRS-Color |
| Prologica | CP-500 | Brasil | TRS-80 Mod.III |
| Ritas | Ringo R-470 | Brasil | Sinclair ZX-81 |
| Sharp | Hotbit HB-8000 | Brasil | MSX |
| Spectrum | Microengenho I | Brasil | Apple II + |
| Spectrum | Microengenho II | Brasil | Apple IIe |
| Spectrum | Spectrum ed | Brasil | Apple II + |
| Suporte | Venus II | Brasil | Apple II + |
| Sycomig | SIC I | Brasil | Apple II + |
| Sysdata | Sysdata III | Brasil | TRS-80 Mod.III |
| Sysdata | Sysdata IV | Brasil | TRS-80 Mod.IV |
| Sysdata | Sysdata Jr. | Brasil | TRS-80 Mod.III |
| Timex | Timex 1000 | USA | Sinclair ZX-81 |
| Timex | Timex 1500 | USA | Sinclair ZX-81 |
| Timex | Timex 2000 | USA | Sinclair Spectrum |
| Unitron | AP II | Brasil | Apple II + |
| Victor do Brasil | Elppa II Plus | Brasil | Apple II + |
| Victor do Brasil | Elppa Jr. | Brasil | Apple II + |
| Video Genie | Video Genie I | USA | TRS-80 Mod. I |

## UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

**INPUT** foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

TRS-Color

Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

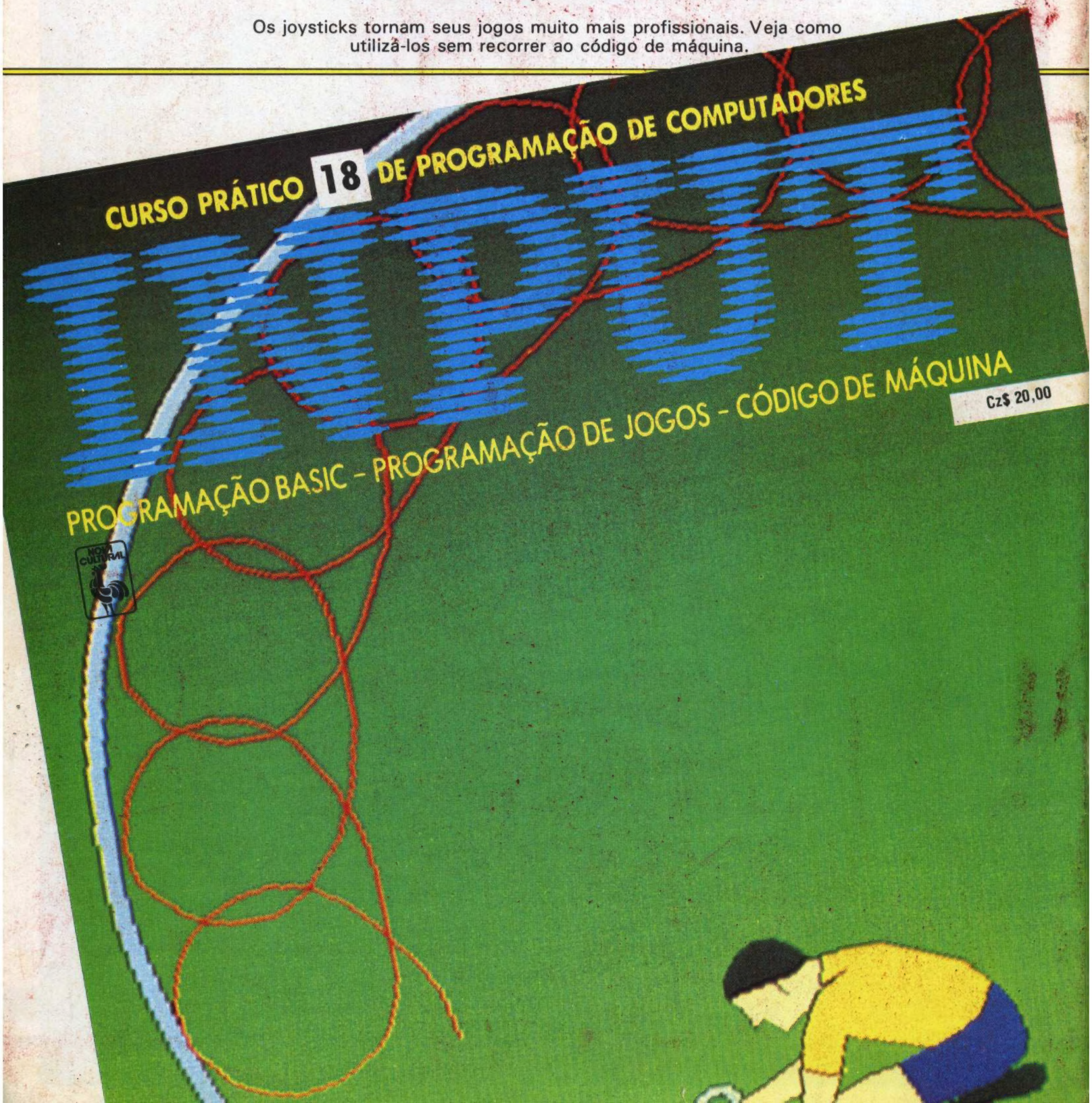
# NO PRÓXIMO NÚMERO

## CÓDIGO DE MÁQUINA

Utilize alguns pequenos programas escritos em código de máquina para dar mais vida a seus programas em BASIC.

## PROGRAMAÇÃO BASIC

Relógios mecânicos e espirais: como elaborar desenhos complexos empregando os comandos **SIN** e **COS**.

## PROGRAMAÇÃO DE JOGOS

Os joysticks tornam seus jogos muito mais profissionais. Veja como utilizá-los sem recorrer ao código de máquina.